

vSphere Command-Line Interface Installation and Reference Guide

ESX/ESXi 4.0 Update 1
vCenter Server 4.0 Update 1

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-000268-00

vmware®

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

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

The VMware Web site also provides the latest product updates.

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

docfeedback@vmware.com

Copyright © 2008, 2009 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

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

Contents

About This Book 7

1 vSphere CLI Installation, Execution, and Command Overviews 9

- Installing and Using vSphere CLI Commands 9
- Installing vSphere CLI and Running Commands on Linux 10
 - Unpacking and Installing the vSphere CLI Package 11
 - Running vSphere CLI Commands 11
 - Uninstalling the vSphere CLI Package 12
- Installing vSphere CLI and Running Commands on Windows 12
 - Uninstalling the vSphere CLI Package 13
- Installing vMA and Running Commands from vMA 13
- Specifying Authentication Information 13
 - Order of Precedence 14
 - Using a Session File 14
 - Using Environment Variables 15
 - Using a Configuration File 15
 - Using Command-Line Options 15
 - Using Microsoft Windows Security Support Provider Interface (SSPI) 16
- Available Options for vSphere CLI Execution 17
- Using vSphere CLI Commands in Scripts 18
- List of Available Commands 19
- Supported Platforms for Commands 20
- Commands with an esxcfg Prefix 23

2 vSphere CLI Reference 25

- Storage Management Commands 25
 - Diagnostic Partition Management with vicfg-dumppart 25
 - Basic Storage Array Management with vicfg-mpath 27
 - NAS File System Management with vicfg-nas 28
 - Available LUN Discovery with vicfg-scsidevs 29
 - Rescan Operations with vicfg-rescan 30
- Networking Commands 31
 - DNS Configuration Specification with vicfg-dns 31
 - Physical NIC Management with vicfg-nics 32
 - NTP Server Specification with vicfg-ntp 33
 - Route Entry Manipulation with vicfg-route 34
 - VMkernel NICs Management with vicfg-vmknics 35
 - Virtual Switch Management with vicfg-vswitch 36
- System Management Commands 39
 - Using resxtp for Performance Monitoring 39
 - Configuration Information Backup with vicfg-cfgbackup 39
 - SNMP Management with vicfg-snmp 40
 - Syslog Server Specification with vicfg-syslog 41
 - Update Management with vihostupdate 42
- Virtual Machine Management Commands 44
 - Virtual Machine Migration with svmotion 44

Miscellaneous Management Commands	46
Advanced Management with vicfg-advcfg	46
VMkernel Module Manipulation with vicfg-module	47
User and Group Management with vicfg-user	48
Volume Management with vicfg-volume	49
File System Manipulation with vifs	50
3 File System Management with vmkfstools	53
vmkfstools Command Syntax	53
Supported Command-Specific Options	54
Supported Targets	55
File System Target	55
Disk Partition Target	55
Device Target	55
vmkfstools File System Options	55
VMFS File System Creation	55
Attribute Listing for a VMFS Volume	56
File System Partition Extension by Spanning	57
vmkfstools Virtual Disk Options	57
Supported Disk Formats	57
Creating Virtual Disks	58
Initializing Virtual Disks	58
Inflating Thin Virtual Disks	59
Deleting Virtual Disks	59
Renaming Virtual Disk	59
Cloning Virtual or Raw Disks	59
Migrating VMware Workstation and VMware GSX Server Virtual Machines	60
Extending Virtual Disks	60
Creating Virtual Compatibility Mode Raw Device Mappings	61
Creating Physical Compatibility Mode Raw Device Mappings	61
Listing RDM Attributes	61
Displaying Virtual Disk Geometry	62
4 iSCSI Storage Management with vicfg-iscsi	63
vicfg-iscsi Command Syntax	63
vicfg-iscsi Options	64
vicfg-iscsi Discovery Target Options	67
Listing Dynamic or Static Discovery Targets	67
Adding Dynamic or Static Discovery Targets	67
Removing Dynamic or Static Discovery Targets	68
vicfg-iscsi Authentication Options	69
Listing Supported and Enabled Authentication Methods	69
Enabling Authentication	69
Enabling Mutual (Bidirectional) Authentication	70
Returning Authentication to Default Inheritance	71
vicfg-iscsi Information Retrieval Options	71
External HBA Properties	71
Target Information	71
LUN Information	72
vicfg-iscsi Physical Network Portal Options	72
Listing Physical Network Portal Information	72
Setting Physical Network Portal Information	72
vicfg-iscsi Network Options	72
Listing Network Properties	73
Setting Network Properties	73

vicfg-iscsi Initiator Options	73
Listing Initiator Properties	73
Setting Initiator iSCSI Name or Alias Name	73
vicfg-iscsi iSCSI Parameter Options	74
Listing iSCSI Parameter Options	74
Listing iSCSI Parameter Option Details	74
Setting iSCSI Parameter Options	74
Returning Parameters to Default Inheritance	75
vicfg-iscsi Software iSCSI Options	75
Listing Software iSCSI Loading Status	75
Enabling Software iSCSI	76
Disabling Software iSCSI	76
Tasks Required for Software iSCSI Setup	76
Tasks Required for Hardware iSCSI Setup	77
5 Virtual Machine Management with vmware-cmd	79
Connection Options for vmware-cmd	79
General Options for vmware-cmd	80
Server Options for vmware-cmd	80
Virtual Machine Options for vmware-cmd	80
Soft and Hard Power Operations	82
vmware-cmd Examples	82
6 Storage Path and Storage Plugin Management with esxcli	83
esxcli Command Syntax	83
Help for esxcli	84
Managing NMP with esxcli nmp	84
Device Management with esxcli nmp device	85
esxcli nmp device list	85
esxcli nmp device setpolicy	85
Fixed Path Selection Policy Operations with esxcli nmp fixed	85
esxcli nmp fixed getpreferred	85
esxcli nmp fixed setpreferred	86
Path Operations with esxcli nmp path	86
esxcli nmp path list	86
PSP Operations with esxcli nmp psp	86
esxcli nmp psp getconfig	86
esxcli nmp psp list	87
esxcli nmp psp setconfig	87
Round Robin Operations with esxcli nmp roundrobin	87
esxcli nmp roundrobin getconfig	87
esxcli nmp roundrobin setconfig	88
SATP Operations with esxcli nmp satp	88
esxcli nmp satp addrule	88
esxcli nmp satp deleterule	89
esxcli nmp satp getconfig	89
esxcli nmp satp list	89
esxcli nmp satp listrules	90
esxcli nmp satp setconfig	90
esxcli nmp satp setdefaultpsp	90

Using esxcli swiscsi nic for NIC Binding	90
esxcli swiscsi nic add	90
esxcli swiscsi nic remove	91
esxcli swiscsi nic list	91
Path Claiming with esxcli corestorage claiming	91
esxcli corestorage claiming reclaim	91
esxcli corestorage claiming unclaim	92
Managing Claim Rules with esxcli corestorage claimrule	92
esxcli corestorage claimrule add	93
esxcli corestorage claimrule convert	94
esxcli corestorage claimrule delete	95
esxcli corestorage claimrule list	95
esxcli corestorage claimrule load	95
esxcli corestorage claimrule move	95
esxcli corestorage claimrule run	96
Path Masking with esxcli corestorage claimrule	96
 Appendix: Commands for Managing Legacy Servers	99
Storage Array Configuration for VMware Infrastructure 3.5 Systems with vicfg-mpath35	99
Performing Maintenance on VMware Infrastructure 3.5 Systems with vihostupdate35	101
 Index	103

About This Book

The *vSphere Command-Line Interface Installation and Reference Guide*, explains how to install and use the VMware® vSphere Command-Line Interface (vSphere CLI) and includes reference documentation and examples for each command.

The *vSphere Command-Line Interface Installation and Reference Guide* covers ESX, ESXi, and vCenter Server.

- [Chapter 1](#) explains how to install vSphere CLI and how to run vSphere CLI commands on Windows or Linux. The chapter also includes a reference to connection options and other options you can use with each of the vSphere CLI commands.
- [Chapter 2](#) provides reference documentation for most of the vSphere CLI commands, organized by subject area. For each command, the chapter lists and discusses each option and gives examples.
- Chapters 3-6 provide reference documentation for more complex vSphere CLI commands.
 - [Chapter 3](#) discusses `vmkfstools`. It includes information about the command syntax, file system options, and virtual disk options.
 - [Chapter 4](#) discusses `vicfg-iscsi`, which you can use for iSCSI monitoring and management operations.
 - [Chapter 5](#) discusses `vmware-cmd`. It includes information about connection options, general options, server options, and virtual machine options.
 - [Chapter 6](#) discusses `esxcli`, which you can use to configure the Native Multipathing Plugin (NMP), the Pluggable Storage Architecture (PSA), and NIC bindings.
- [Appendix A](#) discusses the `vihostupdate35` and `vicfg-mpath35` legacy commands, which are available for ESX/ESXi 3.5 systems.

Intended Audience

This book is for experienced Windows or Linux system administrators who are familiar with vSphere administration tasks and datacenter 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.

Related Documentation

The vSphere documentation consists of the combined VMware vCenter Server and ESX/ESXi documentation set. The documentation set contains background information for the commands discussed in this document.

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 and includes information about the vSphere SDK for Perl Utility Applications.

The *vSphere Management Assistant Guide* explains how to install and use the vSphere Management Assistant (vMA). vMA is a virtual machine that includes the vSphere CLI and other prepackaged software. See [“Installing vMA and Running Commands from vMA”](#) on page 13.

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.vmware.com/support/pubs>.

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.

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 Installation, Execution, and Command Overviews

1

The vSphere CLI command set allows you to run common system administration commands against ESX/ESXi systems from an administration server of your choice. You can run most vSphere CLI commands against a vCenter Server system and target any ESX/ESXi system that system manages. vSphere CLI commands are especially useful for ESXi hosts because they do not include a service console.

vSphere CLI commands run on top of the vSphere SDK for Perl. The vSphere CLI, vSphere SDK for Perl, and some prerequisite software are included in the installation package on Windows and on supported Linux distributions.

This chapter includes the following topics:

- [“Installing and Using vSphere CLI Commands”](#) on page 9
- [“Installing vSphere CLI and Running Commands on Linux”](#) on page 10
- [“Installing vSphere CLI and Running Commands on Windows”](#) on page 12
- [“Installing vMA and Running Commands from vMA”](#) on page 13
- [“Specifying Authentication Information”](#) on page 13
- [“Available Options for vSphere CLI Execution”](#) on page 17
- [“Using vSphere CLI Commands in Scripts”](#) on page 18
- [“List of Available Commands”](#) on page 19
- [“Supported Platforms for Commands”](#) on page 20
- [“Commands with an esxcfg Prefix”](#) on page 23

Installing and Using vSphere CLI Commands

You can install a vSphere CLI package on either Linux or Microsoft Windows, or deploy the vSphere Management Assistant (vMA).

- **vSphere CLI packages** – You can install a vSphere CLI package on a physical or virtual machine, which could become the remote administration server for all ESX/ESXi hosts. See [“Installing vSphere CLI and Running Commands on Linux”](#) on page 10 and [“Installing vSphere CLI and Running Commands on Windows”](#) on page 12.

After you have installed the package, which includes the vSphere SDK for Perl, you can run vSphere CLI commands or run scripts from the operating system command line. Each time you run a command, you specify the connection options directly or indirectly. See [“Specifying Authentication Information”](#) on page 13.

- **vMA** – A virtual machine that administrators can use to run scripts to manage ESX/ESXi servers. vMA includes vSphere CLI and other prepackaged software in a Linux environment.

One important function of vMA is noninteractive login. If you establish an ESX/ESXi host as a target server, you can execute vSphere CLI commands against that server without additional authentication. If you establish a vCenter Server system as a target server, you can execute most vSphere CLI commands against all ESX/ESXi systems it manages without additional authentication. See [“Installing vMA and Running Commands from vMA”](#) on page 13.

You can use vSphere CLI commands interactively or in scripts, as follows:

- Open a command prompt on a Linux or Windows system where you installed the vSphere CLI. Enter commands in that command prompt.
- Access the vMA Linux console. Enter vSphere CLI commands there.
- Prepare scripts with sets of vSphere CLI commands. Then run the scripts from a remote administration server that has the vSphere CLI package installed or from the vMA Linux console. See [“Using vSphere CLI Commands in Scripts”](#) on page 18.

When you run commands, you must specify the ESX/ESXi host against which you want to run the command and the user name and password for authentication. See [“Specifying Authentication Information”](#) on page 13.



CAUTION Specifying the password in plain text risks exposing the password to other users. The password might also become exposed in backup files. Do not provide plain-text passwords on production systems.

You have the following alternatives:

- If you use a vSphere CLI command interactively and do not specify a user name and password, you are prompted for them. The screen does not echo the password you type.
- For noninteractive use, you can create a session file using the `save_session.pl` script included in the `apps/session` directory of the vSphere SDK for Perl. See [“Using a Session File”](#) on page 14.
- If you are running on Windows systems, you can use the `--passthroughauth` command. If the user who runs the command is known, no password is required.
- If you are running vMA, you can set up target servers and execute most vSphere CLI commands against target servers without additional authentication.

Installing vSphere CLI and Running Commands on Linux

The installation script for the vSphere CLI is supported on default installations of the following Linux distributions:

- Red Hat Enterprise Linux (RHEL) 5.2 (64 bit)
- Red Hat Enterprise Linux (RHEL) 5.2 (32 bit)
- SUSE Enterprise Server 10 SP1 32 bit
- Ubuntu 8.04 32 bit

The vSphere CLI package installer installs the vSphere CLI scripts and the vSphere SDK for Perl. If you are not using one of the supported Linux distributions, you must make sure your system has all required libraries installed. See the *vSphere SDK for Perl Installation Guide* for information about required libraries.

If you have uninstalled a previous version of the vSphere CLI (old product name is Remote CLI), and install vSphere CLI in a different directory, you must reset the PATH environment variable. You can do so before or after the installation, using the command appropriate for your distribution or shell (`setenv`, `export`, and so on). If you do not reset the PATH, the system might still go to the old location to find vSphere CLI commands.

Unpacking and Installing the vSphere CLI Package

You must download the vSphere CLI package from the VMware Web site before you can install vSphere CLI.

To download and unpack the vSphere CLI package

- 1 Download the installer package from the VMware Web site.
You can find the installer in the Drivers and Tools tab of the vSphere 4.0 download page.
- 2 Open a shell prompt and navigate to the directory to which you downloaded the package.
- 3 Unzip and unpack the downloaded package.

To install the vSphere CLI package

- 1 Start the installer (`vmware-install.pl`).
- 2 Type **yes** at the prompt to accept the license agreement terms and press Enter to continue.
- 3 When prompted, specify an installation directory, or press Enter to accept the default (`/usr/bin`).
- 4 (Optional) If prompted to remove older versions of vSphere SDK for Perl or vSphere CLI, you can either accept the offer or install the package on a different system.

IMPORTANT The installer replaces both the vSphere SDK for Perl and the vSphere CLI. Install the package on a different system to keep an older version.

The installer indicates completion of the process as follows:

- A success message appears and the prompt returns to the shell prompt.
- The installer lists different version numbers for required modules (if any).
- The vSphere CLI scripts are in the installation directory. The default is `/usr/bin`.

You can run a vSphere CLI command to verify that installation was successful. See [“Running vSphere CLI Commands”](#) on page 11.

vSphere SDK for Perl utility applications and sample scripts that perform common operations on ESX/ESXi hosts and on virtual machines are included with the installation at the following locations:

- Utility applications – `/usr/lib/vmware-vcli/apps`
- Sample scripts – `/usr/share/doc/vmware-vcli/samples`

See the vSphere SDK for Perl documentation for a reference to all utility applications.

Running vSphere CLI Commands

After you have installed the vSphere CLI, you can run commands directly on the Linux command prompt or include them in scripts (see [“Using vSphere CLI Commands in Scripts”](#) on page 18).

To run a vSphere CLI command on Linux

- 1 Open a command prompt.
- 2 Change to the directory where you installed the vSphere CLI. The default is `/usr/bin`.
- 3 Run the command, including the connection options.
`<command> <conn_options> <params>`
Specify connection options in a configuration file or pass them on the command line. The extension `.pl` is not required on Linux.

Example

```
vicfg-nas --server my_esxserver --username u42 --password mypwd --list
```

See [Table 1-2, “vSphere CLI Connection Options,”](#) on page 17 for a complete list of connection options.

Uninstalling the vSphere CLI Package

You can uninstall the vSphere CLI package using a script included in the installation.

To uninstall the vSphere CLI on a Linux system

- 1 Connect to the directory where you installed the vSphere CLI (default is `/usr/bin`).
- 2 Run the `vmware-uninstall-vSphere-CLI.pl` script.

The command uninstalls the vSphere CLI and the vSphere SDK for Perl.

Installing vSphere CLI and Running Commands on Windows

The vSphere CLI installation package for Windows includes the ActivePerl runtime from ActiveState Software and required Perl modules and libraries. The vSphere CLI is supported on the following Windows platforms:

- Windows XP SP2 32 bit
- Windows XP SP2 64 bit
- Windows Vista Enterprise SP1 32 bit
- Windows Vista Enterprise SP1 64 bit

To install the vSphere CLI Package on Windows

- 1 Download the vSphere CLI Windows installer package.
You can find the installer in the Drivers and Tools tab of the vSphere 4.0 download page.
- 2 Start the installer.
- 3 (Optional) If prompted to remove older versions of vSphere SDK for Perl or vSphere CLI, you can either accept the offer or install the package on a different system.

IMPORTANT The installer replaces both the vSphere SDK for Perl and the vSphere CLI. Install this package on a different system to keep an older version.

- 4 Click **Next** in the Welcome page to continue.
- 5 If you do want to install the vSphere CLI in a non-default directory, click **Change** and select the directory.
The default location is `C:\Program Files\VMware\VMware vSphere CLI`.
- 6 Click **Next** to continue.
- 7 Click **Install** to proceed with the installation.

The installation might take several minutes to complete.

After you install the vSphere CLI, you can run commands from the Windows command prompt or include them in scripts (see [“Using vSphere CLI Commands in Scripts”](#) on page 18).

To run a vSphere CLI command on Windows

- 1 Open a command prompt.
- 2 Navigate to the directory in which the vSphere CLI is installed.
`cd C:\Program Files\VMware\VMware vSphere CLI\bin`
- 3 Run the command, passing in the connection options and any other options.
`<command>.pl <conn_options> <params>`

The extension `.pl` is required for most commands, but not for `esxcli`.

Example

```
vicfg-nas.pl --server my_vcserver --username u42 --password mypwd --vihost my_esxhost --list
```

See [Table 1-2, “vSphere CLI Connection Options,”](#) on page 17 for a complete list of connection options.

Uninstalling the vSphere CLI Package

You can uninstall the vSphere CLI package as you would any other package.

To uninstall the vSphere CLI on a Windows system

- 1 Select **Start > Settings > Control Panel > Add or Remove Programs**.
- 2 In the panel that appears, select **vSphere CLI**, and click **Remove**.
- 3 Click **Yes** when prompted.

The system uninstalls both the vSphere SDK for Perl and the vSphere CLI.

Installing vMA and Running Commands from vMA

As an alternative to a package installation, you can deploy vMA on an ESX/ESXi host and run vSphere CLI commands from there. vMA is a virtual machine that administrators can use to run scripts to manage ESX/ESXi systems. vMA includes vSphere CLI and other prepackaged software in a Linux environment.

Setting up vMA consists of a few tasks. The *vSphere Management Assistant Administrator's and Developer's Guide* discusses each task in detail.

- 1 Deploy vMA to an ESX/ESXi system of your choice.

You can use a vSphere Client connected directly to the ESX/ESXi system, or a vSphere Client connected to a vCenter Server system. You can deploy vMA from a URL or download a file, unzip it, and point to the OVF file during the deploy process.

- 2 Configure vMA.

When you boot vMA, it prompts you for the following required configuration information:

- Network information (the default is often acceptable)
- A host name for vMA.
- A password for the vi-admin user. The root user is disabled on vMA.

- 3 (Optional) Add a vCenter Server system or more ESX/ESXi systems to vMA as vi-fastpass targets and call `vifpinit` to initialize vi-fastpass.

After you have specified a host as a vi-fastpass target, specifying connection options is no longer required because authentication has already occurred. You can run vSphere CLI commands against any ESX/ESXi system without specifying connection options for that system explicitly. If you set up a vCenter Server system as a target server, you can specify any of the ESX/ESXi hosts that vCenter Server system manages using the `--vihost` option.

Specifying Authentication Information

When you run a vSphere CLI command, you must specify at a minimum the name of the execution server and the name and password of a user with login privileges. vSphere CLI allows you to run against multiple execution servers from the same administration server, but you must have the correct privileges to perform the actions on each execution server.

Order of Precedence

When you run a vSphere CLI command, authentication happens in the order of precedence presented in [Table 1-1](#).

Table 1-1. vSphere CLI Authentication Precedence

Authentication	Description	See
Command line	Password (<code>--password</code>), session file (<code>--sessionfile</code>), or configuration file (<code>--config</code>) specified on the command line.	“Using a Session File” on page 14
Configuration file	Password specified in a <code>.visdkrc</code> configuration file.	“Using a Configuration File” on page 15
Environment variable	Password specified in an environment variable.	“Using Environment Variables” on page 15
Credential store	Password retrieved from the credential store.	<i>vSphere Web Services SDK Programming Guide</i> and <i>vSphere SDK for Perl Programming Guide</i> .
Current account (Active Directory)	Current account information used to establish an SSPI connection. Available only on Windows.	“Using Microsoft Windows Security Support Provider Interface (SSPI)” on page 16
Prompt the user for a password.	Password is not echoed to screen.	

This order of precedence always applies. That means, for example, that you cannot override an environment variable setting in a configuration file.

Using a Session File

You can create a session file using the `save_session.pl` script. The script is in the `/apps/session` directory of the vSphere SDK for Perl, which is included in the vSphere CLI package. You can use the session file, which does not reveal password information, when you run vSphere CLI commands. If the session file is not used for 30 minutes, it expires.

If you use a session file, any other connection options are ignored.

To create and use a session file

- 1 Connect to the directory where the script is located.

For example:

Windows: `cd C:\Program Files\VMware\VMware vSphere CLI\Perl\apps\session`

Linux: `cd /usr/share/doc/vmware-vcli/samples/session`

- 2 Call `save_session.pl`.

You must specify the server to connect to and the name of a session file in which the script saves an authentication cookie.

```
save_session.pl --savesessionfile <location> --server <server>
```

For example:

Windows: `save_session.pl --savesessionfile C:\Temp\my_session --server my_server`

Linux: `save_session.pl --savesessionfile /tmp/vimsession --server <servername_or_address> --username <username> --password <password>`

If you specify a server, but no user name or password, the script prompts you.

- 3 Pass in the session file using the `--sessionfile` option when you run vSphere CLI commands, as follows:

```
<command> --sessionfile <sessionfile_location> <command_options>
```

For example:

Windows: **vicfg-mpath.pl --sessionfile C:\Temp\my_session --list**

Linux: **vicfg-mpath --sessionfile /tmp/vimsession --list**

Using Environment Variables

You can set environment variables in a Linux profile, in the Environment properties dialog box of the System control panel on Windows, or, for the current session, at the command line. For example:

```
set VI_SERVER=<your_server_name_or_address>
```

IMPORTANT Do not use escape characters in environment variables.

See [“Using vSphere CLI Commands in Scripts”](#) on page 18 for an example.

Using a Configuration File

You can use a text file that contains variable names and settings as a configuration file. Variables corresponding to the options are shown in [Table 1-2, “vSphere CLI Connection Options,”](#) on page 17.



CAUTION Limit read access to a configuration file that contains user credentials.

Pass in the configuration file when you run vSphere CLI commands, as follows:

```
<command> --config <my_saved_config> <option>
```

For example:

```
vicfg-mpath --config <my_saved_config> --list
```

If you have multiple vCenter Server or ESX/ESXi systems and you administer each system individually, you can create multiple configuration files with different names. To run a command or a set of commands on a server, you pass in the `--config` option with the appropriate filename at the command line.

The following example illustrates the contents of a configuration file:

```
VI_SERVER = NN.NNN.NNN.NN
VI_USERNAME = root
VI_PASSWORD = my_password
VI_PROTOCOL = https
VI_PORTNUMBER = 443
```

If you have set up your system to run this file, you can run scripts on the specified server afterwards.

IMPORTANT The password is in plain text, so you must limit access to this file.

Using Command-Line Options

You can pass in command-line options using option name and option value pairs.

```
--<optionname> optionvalue
```

Some options, such as `--help`, have no value.

The following syntax results:

```
<command> --server <vc_server> --username <privileged_user> --password <pw> --vihost <esx_host>
--<option_name option_value>
```

IMPORTANT Enclose passwords and other text with special characters in quotation marks.

On Linux, use single quotes (' '), on Windows, use double quotes (" "). On Linux, you can also use a backslash (\) as an escape character.

Examples

The following examples connect to the server as user `snow-white` with password `dwarf$`. The example calls `vicfg-mpath` with no options. The system displays help information for the command in that case.

The first example (Linux) uses the `\` escape character, the other two use single quotes (Linux) and double quotes (Windows).

Linux

```
vicfg-mpath --server <server> --username snow\white --password dwarf$
vicfg-mpath --server <server> --username 'snow-white' --password 'dwarf$'
```

Windows

```
vicfg-mpath.pl --server <server> --username "snow-white" --password "dwarf$"
```

Using Microsoft Windows Security Support Provider Interface (SSPI)

The `--passthrough` option, which is available if you run vSphere CLI commands from a Microsoft Windows system, allows you to use the Microsoft Windows Security Provider Interface (SSPI). See the Microsoft Web site for a detailed discussion of SSPI.

You can use `--passthroughauth` to establish a connection with a vCenter Server system (vCenter Server system or VirtualCenter Server 3.5 Update 2 or later). After the connection has been established, authentication for the vCenter Server system or any ESX/ESXi system it manages is no longer required. Using `--passthroughauth` passes the credentials of the user who runs the command to the target vCenter Server system. If the user who runs the command is known by both the machine from which you access the vCenter Server system and the machine running the vCenter Server software, no additional authentication is required.

If vSphere CLI commands and the vCenter Server software run on the same machine, a local account for the user who runs the command works. If they run on different machines, the user who runs the command must have an account in a domain trusted by both machines.

SSPI supports a number of protocols. By default, it selects the `Negotiate` protocol, where client and server attempt to find a protocol that both support. You can use `--passthroughauthpackage` to explicitly specify a protocol supported by SSPI. Kerberos, the Windows standard for domain-level authentication, is used frequently. If the vCenter Server system is configured to accept only a specific protocol, specifying the protocol with `--passthroughauthpackage` might be required for successful authentication. If you use `--passthroughauth`, you do not have to specify authentication information in any other way.

Example

```
vicfg-mpath --server <vc_server> --passthroughauth --passthroughauthpackage "Kerberos"
--vihost my_esx --list
```

Connects to a server that has been set up to use SSPI. When you run the command, the system calls `vicfg-mpath` with the `--list` option. The system does not prompt for a user name and password.

Available Options for vSphere CLI Execution

Table 1-2 lists options that are available for all vSphere CLI commands in alphabetical order. The table includes options for use on the command line and variables for use in configuration files.

IMPORTANT For connections, vSphere CLI supports only the IPv4 protocol, not the IPv6 protocol. You can, however, perform IPv6 configuration on the target host with several of the networking commands.

[“Specifying Authentication Information”](#) on page 13 explains how to use the options.

Table 1-2. vSphere CLI Connection Options

Option and Environment Variable	Description
<code>--config <config_file></code> <code>VI_CONFIG=<config_file></code>	Uses the configuration file at the specified location. Specify a path that is readable from the current directory.
<code>--credstore <credstore></code>	Name of a credential store file. Defaults to <code><HOME>/ .vmware/credstore/vicredentials.xml</code> on Linux and <code><APPDATA>/VMware/credstore/vicredentials.xml</code> on Windows. Commands for setting up the credential store are included in the vSphere SDK for Perl, which is installed with the vSphere CLI. The <i>vSphere SDK for Perl Programming Guide</i> explains how to use the credential store.
<code>--encoding <encoding></code> <code>VI_ENCODING=<encoding></code>	Specifies the encoding to be used. One of <code>cp936</code> (Simplified Chinese) <code>ISO-8859-1</code> (German), or <code>Shift_JIS</code> (Japanese). You can use <code>--encoding</code> to specify the encoding the vSphere CLI should map to when it is run on a foreign language system.
<code>--passthroughauth</code> <code>VI_PASSTHROUGHAUTH</code>	If you specify this option, the system uses the Microsoft Windows Security Support Provider Interface (SSPI) for authentication. You are not prompted for a user name and password. See the Microsoft Web site for a detailed discussion of SSPI. This option is supported only if you are running the vSphere CLI on a Windows system and are connecting to a vCenter Server system.
<code>--passthroughauthpackage <package></code> <code>VI_PASSTHROUGHAUTHPACKAGE=<package></code>	Use this option with <code>--passthroughauth</code> to specify a domain-level authentication protocol to be used by Windows. By default, SSPI uses the <code>Negotiate</code> protocol, which means that client and server attempt to negotiate a protocol that both support. If the vCenter Server system to which you are connecting is configured to use a specific protocol, you can specify that protocol using this option. This option is supported only if you are running the vSphere CLI on a Windows system and connecting to a vCenter Server system.
<code>--password <passwd></code> <code>VI_PASSWORD=<passwd></code>	Uses the specified password (used with <code>--username</code>) to log in to the server. <ul style="list-style-type: none"> ■ If <code>--server</code> specifies a vCenter Server system, the user name and password apply to that server. You need no authentication to run commands on the ESX/ESXi hosts that server manages. ■ If <code>--server</code> specifies an ESX/ESXi host, the user name and password apply to that server. Use the empty string (' ' on Linux and " " on Windows) to indicate no password. If you do not specify a user name and password on the command line, the system prompts you and does not echo your input to the screen.
<code>--portnumber <number></code> <code>VI_PORTNUMBER=<number></code>	Uses the specified port to connect to the ESX/ESXi host. Default is 443.
<code>--protocol <HTTP HTTPS></code> <code>VI_PROTOCOL=<HTTP HTTPS></code>	Uses the specified protocol to connect to the ESX/ESXi host. Default is HTTPS.
<code>--savesessionfile <file></code> <code>VI_SAVESESSIONFILE=<file></code>	Saves a session to the specified file. The session expires if it has been unused for thirty minutes.
<code>--server <server></code> <code>VI_SERVER=<server></code>	Uses the specified ESX/ESXi or vCenter Server system. Default is <code>localhost</code> . If <code>--server</code> points to a vCenter Server system, you use the <code>--vihost</code> option to specify the ESX/ESXi host on which you want to run the command. A command is supported for vCenter Server if the <code>--vihost</code> option is defined.

Table 1-2. vSphere CLI Connection Options (Continued)

Option and Environment Variable	Description
--servicepath <path> VI_SERVICEPATH=<path>	Uses the specified service path to connect to the ESX/ESXi host. Default is /sdk/webService.
--sessionfile <file> VI_SESSIONFILE=<file>	Uses the specified session file to load a previously saved session. The session must be unexpired.
--url <url> VI_URL=<url>	Connects to the specified vSphere Web Services SDK URL.
--username <u_name> VI_USERNAME=<u_name>	<p>Uses the specified user name.</p> <ul style="list-style-type: none"> ■ If --server specifies a vCenter Server system, the user name and password apply to that server. You need no authentication to run commands on the ESX/ESXi systems that server manages. ■ If --server specifies an ESX/ESXi system, the user name and password apply to that system. <p>If you do not specify a user name and password on the command line, the system prompts you and does not echo your input to the screen.</p>
--vihost <host> -h <host>	<p>When you run a vSphere CLI command with the --server option pointing to a vCenter Server system, use --vihost to specify the ESX/ESXi host to run the command against.</p> <p>NOTE: This option is not supported for each command. If supported, the option is included in the individual command option list.</p>

Table 1-3 lists options not used as connection options that you can use when you run a vSphere CLI command.

Table 1-3. vSphere CLI Common Options

Option	Description
--help	Prints a brief usage message. The message lists first each command-specific option and then each of the common options.
--verbose	Displays additional debugging information.
--version	Displays version information.
--vihost	<p>When you run a vSphere CLI command with the --server option pointing to a vCenter Server system, use --vihost to specify the ESX/ESXi host to run the command against. Use the host name or IP address, depending on what you specified when you added the host to the vCenter Server system.</p> <p>This option is not supported for all commands and listed in the option table of each command it applies to.</p>

Using vSphere CLI Commands in Scripts

Most administrators run scripts to perform the same task repeatedly or to perform a task on multiple hosts. You can run vSphere CLI commands from one administration server against multiple target servers.

For example, when a new datastore becomes available in your environment, you must make that datastore available to each ESX/ESXi host. The following sample script illustrates how to make a NAS datastore available to three hosts (esxi_server_a, esxi_server_b, and esxi_server_c).

The sample assumes that a configuration file /home/admin/.visdkrc.<hostname> exists for each host. For example, the configuration file for esxi_server_a has the following contents:

```
VI_SERVER = esxi_server_a
VI_USERNAME = root
VI_PASSWORD = xysfdjkat
```

The script itself adds the NAS datastore by calling the different configuration files.

```
#!/bin/sh
for i in {"esxi_server_a","esx_server_b","esxi_server_c"}
do
    echo "Adding NAS datastore for $i..."
    vicfg-nas --config /home/admin/.visdkrc.$i -a -o mainnas.x.com -s /shared nas_ds
    vicfg-nas --config /home/admin/.visdkrc.$i -l
done
```

List of Available Commands

Table 1-4 lists all vSphere CLI commands in alphabetical order and points to the vSphere CLI discussion in this document and related documentation.

Table 1-4. vSphere CLI Commands Supported by ESX/ESXi

Command	Description	See
esxcli	Manage pluggable storage architecture (PSA) and native multipathing (NMP).	“Storage Path and Storage Plugin Management with esxcli” on page 83.
resxtop	Monitors in real time how ESX hosts use resources. Runs in interactive or batch mode. This command is supported only on Linux.	See “Using resxtop for Performance Monitoring” on page 39 for an introduction. See the <i>Resource Management Guide</i> for a detailed reference.
svmotion	Moves a virtual machine’s configuration file and optionally its disks while the virtual machine is running. Must run against a vCenter Server system.	“Virtual Machine Migration with svmotion” on page 44 for an introduction. See the <i>Basic System Administration</i> guide for an in-depth discussion that includes examples.
vicfg-advcfg	Performs advanced configuration including enabling and disabling CIM providers. Use this command as instructed by VMware.	“Advanced Management with vicfg-advcfg” on page 46.
vicfg-cfgbackup	Backs up the configuration data of an ESXi system and restores previously saved configuration data.	“Configuration Information Backup with vicfg-cfgbackup” on page 39. See the <i>ESXi Embedded and vCenter Server Setup Guide</i> for an in-depth discussion that includes step-by-step instructions.
vicfg-dns	Specifies an ESX/ESXi host’s DNS (Domain Name Server) configuration.	“DNS Configuration Specification with vicfg-dns” on page 31.
vicfg-dumppart	Manages diagnostic partitions.	“Diagnostic Partition Management with vicfg-dumppart” on page 25.
vicfg-iscsi	Manages iSCSI storage.	“iSCSI Storage Management with vicfg-iscsi” on page 63.
vicfg-module	Enables VMkernel options. Use this command with the options listed in this document, or as instructed by VMware.	“VMkernel Module Manipulation with vicfg-module” on page 47.
vicfg-mpath vicfg-mpath35	Configures storage arrays. Use <code>vicfg-mpath35</code> for ESX/ESXi 3.5 hosts.	“Basic Storage Array Management with vicfg-mpath” on page 27. “Storage Array Configuration for VMware Infrastructure 3.5 Systems with vicfg-mpath35” on page 99
vicfg-nas	Manages NAS file systems.	“NAS File System Management with vicfg-nas” on page 28.
vicfg-nics	Manages the ESX/ESXi host’s physical NICs.	“Physical NIC Management with vicfg-nics” on page 32.
vicfg-ntp	Specifies the NTP (Network Time Protocol) server.	“NTP Server Specification with vicfg-ntp” on page 33.
vicfg-rescan	Rescans the storage configuration.	“Rescan Operations with vicfg-rescan” on page 30.

Table 1-4. vSphere CLI Commands Supported by ESX/ESXi (Continued)

Command	Description	See
<code>vicfg-route</code>	Manipulates the ESX/ESXi host's route entry.	"Route Entry Manipulation with vicfg-route" on page 34.
<code>vicfg-scsidevs</code>	Finds available LUNs.	"Available LUN Discovery with vicfg-scsidevs" on page 29.
<code>vicfg-snmp</code>	Manages the Simple Network Management Protocol (SNMP) agent.	"SNMP Management with vicfg-snmp" on page 40. Using SNMP in a vSphere environment is discussed in detail in the <i>Basic System Administration</i> manual.
<code>vicfg-syslog</code>	Specifies the syslog server and the port to connect to that server for ESXi hosts.	"Syslog Server Specification with vicfg-syslog" on page 41. The <i>Basic System Administration</i> document discusses system logs in more detail and explains how to set them up using the vSphere Client.
<code>vicfg-user</code>	Creates, modifies, deletes, and lists local direct access users and groups of users.	"User and Group Management with vicfg-user" on page 48.
<code>vicfg-vmknic</code>	Adds, deletes, and modifies virtual network adapters (VMkernel NICs).	"VMkernel NICs Management with vicfg-vmknic" on page 35.
<code>vicfg-volume</code>	Supports resignaturing a VMFS snapshot volume and mounting and unmounting the snapshot volume.	"Volume Management with vicfg-volume" on page 49.
<code>vicfg-vswitch</code>	Adds or removes virtual switches or modifies virtual switch settings.	"Virtual Switch Management with vicfg-vswitch" on page 36.
<code>vifs</code>	Performs file system operations such as retrieving and uploading files on the remote server.	"File System Manipulation with vifs" on page 50.
<code>vihostupdate</code> <code>vihostupdate35</code>	Manages updates of ESX/ESXi hosts. Use <code>vihostupdate35</code> for ESXi 3.5 hosts. See also the <i>ESXi Upgrade Guide</i> .	"Update Management with vihostupdate" on page 42. "Performing Maintenance on VMware Infrastructure 3.5 Systems with vihostupdate35" on page 101
<code>vmkfstools</code>	Creates and manipulates virtual disks, file systems, logical volumes, and physical storage devices on an ESX/ESXi host.	"File System Management with vmkfstools" on page 53.
<code>vmware-cmd</code>	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.	"Virtual Machine Management with vmware-cmd" on page 79.

Supported Platforms for Commands

vSphere CLI 4.0 supports more functionality than vSphere CLI 3.5. Different commands support a different range of target servers.

Most commands can run against an ESX/ESXi system and have vCenter Server support. vCenter Server support means that you can connect to a vCenter Server system, and use `--vihost` to specify the ESX/ESXi host to run the command against. The only exception is `svmotion`, which you can run against vCenter Server systems but not against ESX/ESXi systems.

The following commands must have an ESX/ESXi system, not a vCenter Server system target:

- `vicfg-snmp`
- `vifs`
- `vicfg-user`
- `vicfg-cfgbackup`

- vihostupdate
- vmkfstools
- esxcli

You cannot run the `vihostupdate` and `vicfg-mpath` commands that are in the vSphere CLI 4.0 installation against ESX/ESXi 3.5 or vCenter 2.5 systems. Instead, run `vihostupdate35` and `vicfg-mpath35`, included in the vSphere CLI 4.0 installation, against those systems. `vihostupdate35` is supported for ESXi but not ESX.

IMPORTANT If you run vSphere CLI 4.0 commands against ESX/ESXi 3.5 systems, you can use only the options supported by those systems.

See the *VMware Infrastructure Remote Command-Line Interface Installation and Reference Guide* for ESX/ESXi Update 2 for a list of supported options. To access that document, select Resources, then Documentation from the VMware web site. Find the vSphere documentation set and open the archive. A small number of vSphere CLI 4.0 options are supported against ESX/ESXi 3.5 Update hosts even though they were not supported in RCLI version 3.5.

Run a vSphere CLI 4.0 command with `--help` for information on option support with ESX/ESXi 3.5 Update 2, or see VMware KB article 1008940 for more detail.

[Table 1-5](#) lists platform support for the different vSphere CLI 4.0 commands. These commands have not been tested against VirtualCenter 2.5 Update 2 systems. You can, however, connect to a vCenter Server 4.0 system and target ESX/ESXi 3.5 Update 2 hosts.

Table 1-5. Platform Support for vSphere CLI 4.0 Commands

Command	ESXi 4.0	ESX 4.0	VC 4.0	ESXi 3.5 U2+	ESX 3.5 U2+
<code>esxcli</code>	Yes	Yes	No	No	No
<code>resxtp</code>	Yes	Yes	Yes	Yes	Yes
<code>svmotion</code>	No	No	Yes	No	No
<code>vicfg-advcfg</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-cfgbackup</code>	Yes	No	No	Yes	No
<code>vicfg-dns</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-dumppart</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-iscsi</code>	Yes	Yes	Yes	No	No
<code>vicfg-module</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-mpath</code>	Yes	Yes	Yes	Use <code>vicfg-mpath35</code> instead.	
<code>vicfg-nas</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-nics</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-ntp</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-rescan</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-route</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-scsidevs</code>	Yes	Yes	Yes	No	No
<code>vicfg-snmp</code>	Yes	Yes	No	Yes	Yes
<code>vicfg-syslog</code>	Yes	No	Yes	Yes	No
<code>vicfg-user</code>	Yes	Yes	No	Yes	Yes
<code>vicfg-vmhbadevs</code>	Not included in vSphere CLI 4.0. Use <code>vicfg-scsidevs</code> instead.				
<code>vicfg-vmknics</code>	Yes	Yes	Yes	Yes	Yes
<code>vicfg-volume</code>	Yes	Yes	Yes	No	No
<code>vicfg-vswitch</code>	Yes	Yes	Yes	Yes	Yes
<code>vifs</code>	Yes	Yes	No	Yes	Yes

Table 1-5. Platform Support for vSphere CLI 4.0 Commands (Continued)

Command	ESXi 4.0	ESX 4.0	VC 4.0	ESXi 3.5 U2+	ESX 3.5 U2+
vihostupdate	Yes	Yes	No	Use vihostupdate35 instead	No
vmkfstools	Yes	Yes	No	Yes	Yes
vmware-cmd	Yes	Yes	Yes	Yes	Yes
vicfg-mpath35	No	No	No	Yes	Yes
vihostupdate35	No	No	No	Yes	No

Table 1-6 lists platform support for the different vSphere CLI 3.5 commands. These commands are not supported against vSphere 4.0 systems.

Table 1-6. Platform Support for vSphere CLI 4.0 Commands

Command	ESXi 3.5 U2+	ESX 3.5 U2+	VC 2.5 U2+
esxcli	No	No	No
resxtp	Yes	Yes	No
svmotion	N.A.	N.A.	Yes
vicfg-advcfg	Yes	Yes	Yes
vicfg-cfgbackup	Yes	No	No
vicfg-dns	Yes	Yes	Yes
vicfg-dumppart	Yes	Yes	Yes
vicfg-iscsi	No	No	No
vicfg-module	Yes	Yes	Yes
vicfg-mpath	Yes	Yes	Yes
vicfg-nas	Yes	Yes	Yes
vicfg-nics	Yes	Yes	Yes
vicfg-ntp	Yes	Yes	Yes
vicfg-rescan	Yes	Yes	Yes
vicfg-route	Yes	Yes	Yes
vicfg-scsidevs	No	No	No
vicfg-snmp	Yes	Yes	No
vicfg-syslog	Yes	No	Yes
vicfg-user	Yes	Yes	No
vicfg-vmhbadevs	Yes	Yes	Yes
vicfg-vmknics	Yes	Yes	Yes
vicfg-volume	No	No	No
vicfg-vswitch	Yes	Yes	Yes
vifs	Yes	Yes	No
vihostupdate	Yes	No	No
vmkfstools	Yes	Yes	No
vmware-cmd	Yes	Yes	Yes

Commands with an esxcfg Prefix

For many of the vSphere CLI commands you might have used scripts with corresponding service console commands starting with an `esxcfg` prefix to manage ESX 3.x hosts. To facilitate easy migration from ESX 3.x to ESX/ESXi, the commands with the `esxcfg` prefix are available as vSphere CLI commands.

IMPORTANT VMware recommends that you use the vSphere CLI commands with the `vicfg` prefix. Commands with the `esxcfg` prefix are available mainly for compatibility reasons and might become obsolete.

[Table 1-7](#) lists all vSphere CLI commands for which a command with an `esxcfg` prefix is available.

Table 1-7. Commands with an `esxcfg` Prefix

Command with <code>vicfg</code> prefix	Command with <code>esxcfg</code> prefix	See
<code>vicfg-advcfg</code>	<code>esxcfg-advcfg</code>	“Advanced Management with <code>vicfg-advcfg</code>” on page 46.
<code>vicfg-cfgbackup</code>	<code>esxcfg-cfgbackup</code>	“Configuration Information Backup with <code>vicfg-cfgbackup</code>” on page 39.
<code>vicfg-dns</code>	<code>esxcfg-dns</code>	“DNS Configuration Specification with <code>vicfg-dns</code>” on page 31.
<code>vicfg-dumppart</code>	<code>esxcfg-dumppart</code>	“Diagnostic Partition Management with <code>vicfg-dumppart</code>” on page 25.
<code>vicfg-module</code>	<code>esxcfg-module</code>	“VMkernel Module Manipulation with <code>vicfg-module</code>” on page 47.
<code>vicfg-mpath</code>	<code>esxcfg-mpath</code>	“Basic Storage Array Management with <code>vicfg-mpath</code>” on page 27.
<code>vicfg-nas</code>	<code>esxcfg-nas</code>	“NAS File System Management with <code>vicfg-nas</code>” on page 28.
<code>vicfg-nics</code>	<code>esxcfg-nics</code>	“Physical NIC Management with <code>vicfg-nics</code>” on page 32.
<code>vicfg-ntp</code>	<code>esxcfg-ntp</code>	“NTP Server Specification with <code>vicfg-ntp</code>” on page 33.
<code>vicfg-rescan</code>	<code>esxcfg-rescan</code>	“Rescan Operations with <code>vicfg-rescan</code>” on page 30.
<code>vicfg-route</code>	<code>esxcfg-route</code>	“Route Entry Manipulation with <code>vicfg-route</code>” on page 34.
<code>vicfg-scsidevs</code>	<code>esxcfg-scsidevs</code>	“Available LUN Discovery with <code>vicfg-scsidevs</code>” on page 29.
<code>vicfg-snmp</code>	<code>esxcfg-snmp</code>	“SNMP Management with <code>vicfg-snmp</code>” on page 40.
<code>vicfg-syslog</code>	<code>esxcfg-syslog</code>	“Syslog Server Specification with <code>vicfg-syslog</code>” on page 41.
<code>vicfg-vmknic</code>	<code>esxcfg-vmknic</code>	“VMkernel NICs Management with <code>vicfg-vmknic</code>” on page 35.
<code>vicfg-vswitch</code>	<code>esxcfg-vswitch</code>	“Virtual Switch Management with <code>vicfg-vswitch</code>” on page 36.

vSphere CLI Reference

vSphere CLI includes commands for storage, network, system, and virtual machine management discussed here. Some commands with many options are discussed in separate chapters.

This chapter includes the following topics:

- [“Storage Management Commands”](#) on page 25
- [“Networking Commands”](#) on page 31
- [“System Management Commands”](#) on page 39
- [“Virtual Machine Management Commands”](#) on page 44
- [“Miscellaneous Management Commands”](#) on page 46
- [“File System Manipulation with vifs”](#) on page 50

Storage Management Commands

[Table 2-1](#) lists the vSphere CLI storage management commands.

Table 2-1. Storage Management Commands

Command	See
esxcli	Chapter 6, “Storage Path and Storage Plugin Management with esxcli,” on page 83.
vicfg-dumppart	“Diagnostic Partition Management with vicfg-dumppart” on page 25.
vicfg-iscsi	Chapter 4, “iSCSI Storage Management with vicfg-iscsi,” on page 63.
vicfg-mpath	“Basic Storage Array Management with vicfg-mpath” on page 27.
vicfg-nas	“NAS File System Management with vicfg-nas” on page 28.
vicfg-rescan	“Rescan Operations with vicfg-rescan” on page 30.
vicfg-scsidevs	“Available LUN Discovery with vicfg-scsidevs” on page 29.
vmkfstools	.Chapter 3, “File System Management with vmkfstools,” on page 53

Diagnostic Partition Management with vicfg-dumppart

The `vicfg-dumppart` command queries, sets, and scans an ESX/ESXi host’s diagnostic partitions. The *ESX Configuration Guide* and the *ESXi Configuration Guide* discuss diagnostic partitions in detail.

Options

Table 2-2 lists the command-specific options for `vicfg-dumppart`. You must also specify connection options. See “[vSphere CLI Connection Options](#)” on page 17.

Table 2-2. Options for `vicfg-dumppart`

Option	Description
<code>--activate <partition></code> <code>-a <partition></code>	Makes the specified partition the current diagnostic partition. This option has the same effect as <code>--set</code> .
<code>--deactivate [<partition>]</code> <code>-d [<partition>]</code>	Deactivates the active diagnostic partition. The option also unsets the diagnostic partition. Specifying the name of the partition is optional. CAUTION: If you run <code>vicfg-dumppart</code> with this option, your system cannot write errors to a file until another partition is activated. You lose any error record if errors occur.
<code>--find</code> <code>-f</code>	Finds all diagnostic partitions on the ESX/ESXi host and prints them. The partitions can include, in order of suitability, parallel adapter, block adapter, Fibre Channel, hardware iSCSI, software iSCSI partitions.
<code>--get-active</code> <code>-t</code>	Displays the active diagnostic partition for this system. Running <code>vicfg-dumppart</code> with this option returns the internal name of the partition (<code>vmhbaX:X:X:X</code>) or none if no partition is set.
<code>--list</code> <code>-l</code>	Lists all partitions on the ESX/ESXi system that have the appropriate partition type to act as a diagnostic partition. CAUTION: Execution might take several minutes and slow down your ESX/ESXi host because the command scans all LUNs on the system.
<code>--set vmhba<partition></code> <code>-s vmhba<partition></code>	Sets and activates the diagnostic partition, which you must specify using <code>naa.xxx:1</code> or <code>eui.xxx</code> syntax.
<code>--vihost <host></code> <code>-h <host></code>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.

Examples

The following examples for `vicfg-dumppart` require connection options, indicated by `<conn_options>`. See “[Specifying Authentication Information](#)” on page 13.

```
vicfg-dumppart.pl <conn_options> -t
```

Shows the current diagnostic partition the VMkernel uses.

```
vicfg-dumppart.pl <conn_options> -s naa.<naa_ID>
```

Sets the active partition to `naa.<naa_ID>`. After the command has been run, `-t` reports `naa.<naa_ID>` as the active partition.

```
vicfg-dumppart.pl <conn_options> -l
```

Displays information on all partitions that can be used as diagnostic partitions.

```
vicfg-dumppart.pl <conn_options> -d
```

Deactivates the diagnostic partition. After this command has run, no diagnostic partition is set.

```
vicfg-dumppart.pl <conn_options> -f
```

Finds all partitions that could be used as diagnostic partitions and displays detailed information. The output might look as follows.

```
Partition name on vml.mpx.vmhba36:C0:T0:L0:7 -> mpx.vmhba36:C0:T0:L0:7
```

Basic Storage Array Management with vicfg-mpath

The `vicfg-mpath` command supports listing information about Fibre Channel or iSCSI LUNs and changing a path's state. Use the `esxcli` command for managing pluggable storage architecture (PSA) and native multipathing (NMP), including path policy modification. See [“Storage Path and Storage Plugin Management with esxcli”](#) on page 83.

IMPORTANT This command is available only for ESX/ESXi 4.0 or later. Use the `vicfg-mpath35` command to perform storage management on ESX/ESXi 3.5 hosts. See [“Storage Array Configuration for VMware Infrastructure 3.5 Systems with vicfg-mpath35”](#) on page 99.

Names of virtual machine HBAs are not guaranteed to be valid across reboots. Use industry-standard device names, with format `eu i . xxx` and `naa.xxx` to be sure of consistency. Do not use VML LUN names unless device names are not available.

Options

Table 2-3 lists the command-specific options for `vicfg-mpath`. You must also specify connection options. See Table 1-2, [“vSphere CLI Connection Options,”](#) on page 17.

Table 2-3. Options for `vicfg-mpath`

Option	Description
<code>--device</code> <code>-d</code>	Used with the <code>--list</code> option to display only a specific device.
<code>--list [-P -d]</code> <code>-l [-P -d]</code>	Lists detailed path information for the specified path or device.
<code>--list-compact [-P -d]</code> <code>-L [-P -d]</code>	Presents a short listing of all paths.
<code>--list-map [-P -d]</code> <code>-m [-P -d]</code>	Lists all paths and the corresponding adapters and device mappings.
<code>--list-paths [-P -d]</code> <code>-b [-P -d]</code>	Lists all devices and the corresponding paths.
<code>--list-plugins [-P -d]</code> <code>-G [-P -d]</code>	Lists all multipathing plugins loaded into the system. At a minimum, this command returns NMP (Native Multipathing Plugin). If other MPP plugins have been loaded, they are listed as well. For information about storage array plugins, see the <i>ESX Configuration Guide</i> and the <i>ESXi Configuration Guide</i> . See Chapter 6, “Storage Path and Storage Plugin Management with esxcli,” on page 83 for plugin management commands.
<code>--path</code> <code>-P</code>	Used with list commands to display only information for a specific path. The path name can be the long path UID or the shorter runtime name of the path.
<code>--state [active off]</code> <code>-s [active off]</code>	Sets the state of a given LUN path to either <code>active</code> or <code>off</code> . This option requires that the <code>--path</code> options is set and specifies either the path UID or the path runtime name. If you are changing a path's state: <ul style="list-style-type: none"> ■ 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.
<code>--vhost <host></code> <code>-h <host></code>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vhost</code> to specify the ESX/ESXi host to run the command against.

Examples

The following examples for `vicfg-mpath` require connection options. The server must be an ESX/ESXi 4.x host. See [“Specifying Authentication Information”](#) on page 13.

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

Sets the state for a specific path. Requires that `--path` is specified.

vicfg-mpath.pl <conn_options> -m

Lists all paths with adapter and device mappings.

vicfg-mpath.pl <conn_options> -l

Lists all paths on the system with their detailed information.

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

Lists detailed information about the path specified by -P. You can specify the path using the runtime name or the first item in the `vicfg-mpath -l` display. See the preceding example for `vicfg-mpath -l`.

vicfg-mpath.pl <conn_options> -l -d mpx.vmhba32:C0:T1:L0

Lists detailed information for the paths for the device specified in -d.

NAS File System Management with vicfg-nas

The `vicfg-nas` command manipulates NAS file systems associated with ESX/ESXi systems. For more information on working with NAS file systems, see the *ESX Configuration Guide* and the *ESXi Configuration Guide*.

Options

Table 2-4 lists command-specific options for `vicfg-nas`. You must also specify connection options. See “[vSphere CLI Connection Options](#)” on page 17.

Table 2-4. Options for `vicfg-nas`

Option	Description
<code>--add <name></code> <code>-a <name></code>	Adds a new NAS file system to the ESX/ESXi host. When you use this option, you must also use the <code>-o</code> and <code>-s</code> options, and must specify a label name for the new file system.
<code>--delete <name></code> <code>-d <name></code>	Deletes a NAS file system. This command unmounts the NAS file system and removes it from the list of known file systems.
<code>--list</code> <code>-l</code>	Lists all known NAS file systems with their mount name, share name, and host name and indicates for each file system whether it is mounted.
<code>--nasserver <n_host></code> <code>-o <n_host></code>	Used with the <code>-a</code> option to supply the host name for a new NAS file system.
<code>--readonly</code> <code>-y</code>	Adds the new NAS file system with read-only access.
<code>--share <share></code> <code>-s <share></code>	Used with the <code>-a</code> option to supply the share name for a new NAS file system.
<code>--vihost <host></code> <code>-h <host></code>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.

Examples

The following examples for `vicfg-nas` require connection options. See “[Specifying Authentication Information](#)” on page 13.

vicfg-nas.pl <conn_options> -l

Lists all known NAS file systems with their mount name, share name, and host name. Indicates for each file system whether it is mounted.

vicfg-nas.pl <conn_options> -a

Adds a new NAS file system to the ESX/ESXi host. This command adds an entry to the known NAS file system list.

vicfg-nas.pl <conn_options> -d

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

vicfg-nas.pl <conn_options> -a -s new_share

Adds a new NAS file system to the ESX/ESXi host. This command adds an entry to the known NAS file system list and supplies the share name of the new NAS file system.

Available LUN Discovery with vicfg-scsidevs

The **vicfg-scsidevs** command displays information about available LUNs on ESX/ESXi 4.0 hosts.

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

In previous releases of this command-line interface, the corresponding command is **vicfg-vmhbadevs**.

Options

You can run **vicfg-scsidevs** with the following options. For additional options, see [“vSphere CLI Connection Options”](#) on page 17.

Table 2-5. Options for vicfg-scsidevs

Option	Description
--compact-list -c	Lists all logical devices, each on a single line, with limited information. The information includes the device ID, device type, size, and plugin and device display name. You can specify the --device option to list information about a specific device.
--device <device> -d <device>	Used with other options to specify the device for which you want information.
--hba-device-list -A	For each HBA, prints a mapping between the HBA and the devices for which it provides paths.
--hbas -a	Prints HBA devices with identifying information. This includes the adapter ID, driver ID, adapter UID, PCI, vendor, and model.
--list -l	Lists device information for all logical devices on this system. The information includes the name (UUID), device type, display name, and multipathing plugin. You can specify the --device option to list information about a specific device.
--query -q	Prints mappings in 2.5 compatibility mode to mimic a call to vmkpcidivv -q vmhba_devs .
--uids -u	Lists the primary UID for each device and any other UIDs (aliases) for each UID. You can specify the --device option to list information about a specific device.
--vihost <host> -h <host>	When you run a vSphere CLI command with the --server option pointing to a vCenter Server system, use --vihost to specify the ESX/ESXi host to run the command against.
--vmfs -m	Prints mappings for each VMFS volume to its corresponding partition, path to that partition, VMFS UUID, extent number and volume names.

Examples

The following examples for **vicfg-scsidevs** require connection options. See [“Specifying Authentication Information”](#) on page 13.

vicfg-scsidevs.pl <conn_options> -l

Lists detailed information about all logical devices on this system. Here is the output for just one device; the actual listing includes multiple devices.

```
mpx.vmhba2:C0:T1:L0
Device Type: cdrom
Size: 0 MB
Display Name: Local SONY CD-ROM (mpx.vmhba2:C0:T1:L0)
Plugin: NMP
```

```

Console Device: /vmfs/devices/genscsi/mpx.vmhba2:C0:T1:L0
Devfs Path: /vmfs/devices/genscsi/mpx.vmhba2:C0:T1:L0
Vendor: SONY      Model: DVD RW AW-Q170A   Revis: 1.70
SCSI Level: 5   Is Pseudo:   Status:
Is RDM Capable: Is Removable:
Other Names: vml.000N000000NNNdNNNNNNNNaNNNaNN

```

vicfg-scsidevs.pl <conn_options> -c

Lists abbreviated information about all logical devices on this system.

vicfg-scsidevs.pl <conn_options> -u

Lists all device unique identifiers with their primary name.

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

Lists detailed information for a specific logical device.

vicfg-scsidevs.pl <conn_options> -m

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

vicfg-scsidevs.pl <conn_options> -a

Prints HBA devices with identifying information.

vicfg-scsidevs.pl <conn_options> -A

Prints a mapping between HBAs and the devices they provide paths to.

Rescan Operations with vicfg-rescan

You can use `vicfg-rescan` or the vSphere Client to perform a rescan. Perform a rescan operation each time you reconfigure your storage setup. See the *ESX Configuration Guide* and the *ESXi Configuration Guide*. The *Fibre Channel SAN Configuration Guide* discusses rescan on Fibre Channel storage. The *iSCSI SAN Configuration Guide* discusses rescan on iSCSI storage.

When you rescan an ESX/ESXi host, the command returns only an indication of success or failure and no detailed information.

Options

Table 2-6 lists the command-specific options for `vicfg-rescan`. You must also specify connection options. See “vSphere CLI Connection Options” on page 17.

Table 2-6. Options for `vicfg-rescan`

Option	Description
<vmkernel_SCSI_adapter_name>	Name of the adapter to scan, for example, <code>vmhba0</code> .
--vihost <host> -h <host>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.

Example

The following example for `vicfg-rescan` requires connection options. See Table 1-2, “vSphere CLI Connection Options,” on page 17.

```
vicfg-rescan.pl <conn_options> vmhba0
```

Scans `vmhba0` and results in output like the following:

```
Scan operation succeeded
```

Networking Commands

[Table 2-7](#) lists vSphere CLI networking commands. For an introduction to vSphere networking, see the networking chapters in the *ESX Configuration Guide* and the *ESXi Configuration Guide*.

Table 2-7. vSphere CLI Networking Commands

Command	See
<code>vicfg-dns</code>	“DNS Configuration Specification with vicfg-dns” on page 31
<code>vicfg-nics</code>	“Physical NIC Management with vicfg-nics” on page 32
<code>vicfg-ntp</code>	“NTP Server Specification with vicfg-ntp” on page 33
<code>vicfg-route</code>	“Route Entry Manipulation with vicfg-route” on page 34
<code>vicfg-vmknic</code>	“VMkernel NICs Management with vicfg-vmknic” on page 35
<code>vicfg-vswitch</code>	“Virtual Switch Management with vicfg-vswitch” on page 36
<code>esxcli swiscsi nic</code>	“Using esxcli swiscsi nic for NIC Binding” on page 90

DNS Configuration Specification with vicfg-dns

The `vicfg-dns` command lists and specifies the DNS configuration of your ESX/ESXi host. Call the command without command-specific options to list the existing DNS configuration.

If you attempt to change the host or domain name or the DNS server on hosts that use DHCP (dynamic host protocol), an error results.

Options

[Table 2-8](#) lists the command-specific options for `vicfg-dns`. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-8. Options for `vicfg-dns`

Option	Description
<code>--dhcp yes no</code> <code>-H yes no</code>	Specifies whether or not the ESX/ESXi host should use DHCP to determine DNS configuration. You must specify <code>yes</code> or <code>no</code> .
<code>--dns <server_list></code> <code>-D <server_list></code>	DNS server or servers to be used. Specify a comma-separated list of DNS servers, in order of preference.
<code>--domain <d_name></code> <code>-d <d_name></code>	Domain name portion for this ESX/ESXi host. For example, <code>comp-xyz.com</code> .
<code>--hostname <h_name></code> <code>-n <h_name></code>	Human-readable host name portion of the DNS name. For example, <code>esx01</code> .
<code>--vihost</code> <code>-h</code>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.
<code>--vnic <v_nic></code> <code>-V <v_nic></code>	The virtual network adapter to use when overriding the system DNS. This option is required when <code>--dhcp</code> is <code>yes</code> . For ESX hosts, <code>v_nic</code> must be one of the service console network adapters. For ESXi hosts, <code>v_nic</code> must be one of the VMkernel network adapter. The specified virtual network adapter must have DHCP configured.

Examples

The following examples for `vicfg-dns` require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-dns.pl <conn_options> -n dns_host_name
```

Configures the host name for the specified server.

vicfg-dns.pl <conn_options> -d mydomain.biz

Configures the domain name for the specified server.

vicfg-dns.pl <conn_options> -H yes

Specifies that the host should use DHCP to determine the DNS configuration.

vicfg-dns.pl <conn_options>

Displays DNS properties for the specified server. The information includes the host name, domain name, DHCP setting (true or false), and DNS servers on the ESX/ESXi host. For example:

```
Host Name      localhost
Domain Name    localdomain
DHCP           true
DNS Servers    10.NN.NNN.1
               10.NN.NNN.2
```

Physical NIC Management with vicfg-nics

The **vicfg-nics** command manages physical NICs (uplink adapters), that is, the Ethernet switches used by an ESX/ESXi host.

You can use **vicfg-nics** to list information and to specify speed and duplex setting for a NIC.

Options

[Table 2-9](#) lists the command-specific options for **vicfg-nics**. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-9. Options for vicfg-nics

Option	Description
--auto -a	Sets the specified network adapter to auto-negotiate its speed and duplex settings.
--duplex [full half] <nic> -d [full half] <nic>	Sets the duplex value at which a given network adapter should run to either full (transmit data in both directions at the same time) or half (transmit data in one direction at a time).
--list -l	Lists the network adapter's VMkernel name, its PCI ID, driver, link state, speed, duplex setting, and a short PCI description of the card.
--speed <speed> <nic> -s <speed> <nic>	Sets the speed at which a given network adapter should run. Valid values for <speed> are 10, 100, 1000, or 10000.
--vihost <host> -h <host>	When you run a vSphere CLI command with the --server option pointing to a vCenter Server system, use --vihost to specify the ESX/ESXi host to run the command against.

Examples

The following examples for **vicfg-nics** require connection options. See [“Specifying Authentication Information”](#) on page 13.

vicfg-nics.pl <conn_options> -d full -s 100 vmnic0

Sets the duplex setting of **vmnic0** to **full** and the speed to **100**.

vicfg-nics.pl <conn_options> -a vmknic2

Sets **vmknic2** to auto-negotiate its speed and duplex settings.

vicfg-nics.pl <conn_options> -l

Lists the NICs in the system, and prints their current and configured speed and duplex setting.

The system returns a table that includes the name, PCI, driver, link speed, duplex MTU, and description information, for example:

Name	PCI	Driver	Link	Speed	Duplex	MTU	Description
vmnic1	02:02.0 e1000	<driver>	Up	1000Mbps	Full	1500	<desc>
vmnic2	02:04.0 e1000	<driver>	Down				<desc>
vmnic0	00:02.0 e1000	<driver>	Up	1000Mbps	Full	1500	<desc>

NTP Server Specification with vicfg-ntp

The `vicfg-ntp` command supports specifying the NTP (Network Time Protocol) server for an ESX/ESXi host. Some protocols, such as Kerberos, must have accurate information about the current time.

NTP setup requires these calls to `vicfg-ntp`.

- 1 A call to `vicfg-ntp --add` to add the NTP server.
- 2 A call to `vicfg-ntp --start` to start the service.
- 3 A call to `vicfg-ntp --list` to list the service.

Options

[Table 2-10](#) lists the command-specific options for `vicfg-ntp`. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-10. Option for `vicfg-ntp`

Option	Description
<code>--add <ntp_server></code> <code>-a <ntp_server></code>	Adds the NTP server that the host name or IP address specifies.
<code>--delete <ntp_server></code> <code>-d <ntp_server></code>	Deletes the NTP server that the host name or IP address specifies.
<code>--list</code> <code>-l</code>	Displays a list of all NTP servers used by this host.
<code>--start</code> <code>-r</code>	Starts the NTP service on the host.
<code>--stop</code> <code>-s</code>	Stops the NTP service on the host.
<code>--vihost <host></code> <code>-h <host></code>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.

Examples

The following examples for `vicfg-ntp` require connection options. See [“Specifying Authentication Information”](#) on page 13.

vicfg-ntp.pl <conn_options> -l

Displays a list of NTP servers used by this host, or No NTP Servers configured.

vicfg-ntp.pl <conn_options> --start

Starts the NTP service on the specified host.

vicfg-ntp.pl <conn_options> -a 192.XXX.XXX.XX

Adds the host specified by 192.XXX.XXX.XX as a remote NTP server.

Route Entry Manipulation with `vicfg-route`

The `vicfg-route` command lists or sets the default IP gateway. Changing the gateway might be required if you move your ESX/ESXi host to a new physical location. The `vicfg-route` command supports a subset of the Linux `route` command's options.

If you run `vicfg-route` with no options, the command displays the default gateway. You can use `--family` to print the default IPv4 or the default IPv6 gateway. By default, the command displays the default IPv4 gateway.

Options

Table 2-11 lists the command-specific options for `vicfg-route`. You must also specify connection options. See “[vSphere CLI Connection Options](#)” on page 17.

Table 2-11. Options for `vicfg-route`

Option	Description
<gateway>	The IP address or the host name of the machine that should be set as the gateway for the VMkernel IP stack.
--add <route> -a <route>	Adds a route entry to the VMkernel. Use <code>--add <route> default</code> to add a route entry and make it the default.
--delete <route> -d <route>	Deletes a route entry from the VMkernel.
--list -l	Lists route information.
--family v4 v6 -f v4 v6	Address family, either v4 for IPv4 or v6 for IPv6. Defaults to v4.
--vihost <host> -h <host>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.

Examples

The following examples for `vicfg-route` require connection options. See “[Specifying Authentication Information](#)” on page 13.

```
vicfg-route.pl --server <server name> --username <user name> --password <password> <new_gateway>
```

Sets the IP stack of the VMkernel default gateway entry, changing the default gateway to a different gateway.

```
vicfg-route.pl <conn_options> -a 192.NNN.100.0/24 192.NNN.0.1
```

```
vicfg-route.pl <conn_options> -a 192.NNN.100.0 255.255.255.0 192.NNN.0.1
```

Adds a route entry to 192.NNN.100.0 network through 192.NNN.0.1

```
vicfg-route.pl <conn_options> 192.NNN.0.1
```

```
vicfg-route.pl <conn_options> -a default 192.NNN.0.1
```

Sets the VMkernel default gateway to 192.NNN.0.1

```
vicfg-route.pl <conn_options> -d 192.NNN.100.0/24 192.NNN.0.1
```

Deletes the 192.NNN.100.0 route entry from the VMkernel.

```
vicfg-route.pl <conn_options> -f V6 -a 2001:NN:NN:253::/64 2001:NN:NN:253::1
```

Adds a route to 2001:NN:NN:253::/64 network through 2001:NN:NN:253::1

```
vicfg-route.pl <conn_options> -f V6 -a default 2001:NN:NN:253::1
```

Sets the VMkernel default gateway to 2001:NN:NN:253::1.

```
vicfg-route.pl <conn_options> -f V6 -d 2001:NN:NN:253::/64 2001:NN:NN:253::1
```

Deletes a 2001:NN:NN:253:: route entry from the VMkernel.

VMkernel NICs Management with vicfg-vmknic

The `vicfg-vmknic` command configures VMkernel NICs (virtual network adapters).

The `<port_group>` argument used with the `--delete` and `--enable` options specifies the port group the VMkernel NIC is associated with.

Use the `esxcli swisis nic` command to specify NIC bindings for VMkernel NICs. See [“Using esxcli swisis nic for NIC Binding”](#) on page 90.

Options

Table 2-12 lists the command-specific options for `vicfg-vmknic`. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-12. Options for `vicfg-vmknic`

Option	Description
<code><port_group></code>	Specifies the port group. The name is the same as the VMkernel NIC name. Required argument.
<code>--add</code> <code>-a</code>	Adds a VMkernel NIC to the system. You must specify the IP address using <code>--ip</code> , the netmask, and the port group name. When the command completes successfully, the newly added VMkernel NIC is enabled. You cannot specify the <code>dvsName</code> and <code>dvportId</code> parameters with this option.
<code>--delete <port_group></code> <code>-d <port_group></code>	Deletes the VMkernel NIC on the specified port group.
<code>--disable-vmotion <port_group></code> <code>-u <port_group></code>	Disables VMotion for the VMkernel NIC on a specified port group.
<code>--dvs-name</code> <code>-s</code>	Specifies the distributed virtual switch (DVS) name to use as a connection point. DVS is also known as vNetwork Distributed Switch (VDS). Requires that <code>--dvport-id</code> is also specified.
<code>--dvport-id</code> <code>-v</code>	Specifies the distributed virtual port ID of the connection point. Requires that <code>--dvs-name</code> is also specified.
<code>--enable-ipv6 true false</code> <code>-6 true false</code>	Enables or disables IPv6 for this VMkernel NIC for the next boot.
<code>--enable-vmotion <port_group></code> <code>-E <port_group></code>	Enables VMotion for the VMkernel NIC on a specified port group. By default, VMotion is disabled for a port group. To perform migration with VMotion over the network, you have to enable VMotion.
<code>--ip <ip_address> DHCP</code> <code>-i <ip_address> DHCP</code>	Optional. Sets the IP address for this VMkernel NIC. The IP address can have one of the following formats: <ul style="list-style-type: none"> ■ <code><X.X.X.X></code> – Use the specified static IPv4 address. ■ <code>DHCP</code> – Use IPv4 DHCP. ■ <code><X:X:X::/X></code> – Use the specified static IPv6 address ■ <code>DHCPV6</code> – Enable the IPv6 DHCP address ■ <code>AUTOCONF</code> – Enable the IPv6 address advertised by the router If you specify DHCP, the VMkernel must support DHCP.
<code>--list</code> <code>-l</code>	Lists virtual network adapters on the system. The list contains the network information, port group, MTU, and current state for each virtual network adapter in the system.
<code>--mtu <mtu></code> <code>-m <mtu></code>	Maximum transmission unit for the interface being created.
<code>--netmask <netmask></code> <code>-n <netmask></code>	IP netmask (X.X.X.X) to be used for the virtual network adapter. When you set a netmask, you must specify the <code>--ip</code> option in the same command.
<code>--portgroup <group></code> <code>-P <group></code>	Specifies the port group to use as a connection point.
<code>--tso</code> <code>-t</code>	Disables TCP Segment Offloading (TSO) for the VMkernel NIC being created.

Table 2-12. Options for `vicfg-vmknic` (Continued)

Option	Description
<code>--unset-ip <ip_addr></code> <code>-U <ip_addr></code>	Unsets the IP address for this VMkernel NIC. The address can have one of the following formats: <ul style="list-style-type: none"> ■ <code><X:X:X::X></code> – Use the specified static IPv6 address ■ DHCPV6 – Enable IPv6 DHCP address ■ AUTOCONF – Enable the IPv6 address advertised by the router
<code>--vihost <host></code> <code>-h <host></code>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.

Examples

The following examples for `vicfg-vmknic` require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-vmknic <conn_options> -a --ip <IP address> -n <net mask> "VMkernel NIC Name"
```

Adds a VMkernel NIC to the system.

```
vicfg-vmknic <conn_options> -d "VMkernel NIC Name"
```

Deletes a NIC.

```
vicfg-vmknic <conn_options> -d --dvs-name "dvs-001" --dvport-id 1
```

Deletes a NIC.

```
vicfg-vmknic <conn_options> -l
```

Lists all the VMkernel NICs.

```
vicfg-vmknic <conn_options> --ip <IP address> -n <net mask> "VMkernel NIC Name"
```

Modifies a NIC's IP address and netmask.

```
vicfg-vmknic <conn_options>--ip DHCP "VMkernel NIC Name"
```

Sets the NIC to use DHCP.

```
vicfg-vmknic <conn_options>--enable-ipv6 true
```

Enables IPv6 for next boot.

```
vicfg-vmknic <conn_options>--enable-vmotion "portgroup name"
```

Enables VMotion for the VMkernel NIC on a specified port group.

```
vicfg-vmknic <conn_options> --disable-vmotion "portgroup name"
```

Disables VMotion for the VMkernel NIC on a specified port group.

Virtual Switch Management with `vicfg-vswitch`

The `vicfg-vswitch` command adds or removes virtual switches or modifies virtual switch settings. A virtual switch is an abstracted network device. It can route traffic internally between virtual machines and link to external networks. The *ESX Configuration Guide* and the *ESXi Configuration Guide* discuss virtual switches, distributed virtual switches (DVS/VDS), port groups, and DVS/VDS port groups.

By default, each ESX/ESXi host has a single virtual switch called `vSwitch0`.

Options

Table 2-13 lists the command-specific options for `vicfg-vswitch`. You must also specify connection options. See “vSphere CLI Connection Options” on page 17.

Table 2-13. Options for `vicfg-vswitch`

Option	Description
<code>--add <vswitch_name></code> <code>-a <vswitch_name></code>	Adds the specified virtual switch to the system.
<code>--add-pg <portgroup> <vswitch_name></code> <code>-A <portgroup> <vswitch_name></code>	Adds a port group to the specified virtual switch.
<code>--add-dvp-uplink <adapter_name></code> <code>--dvp <DVPort_id> <dvs switch_name></code> <code>-P <adapter_name> -V <DVPort_id></code> <code><dvs switch_name></code>	Adds an uplink adapter (physical network adapter) to a distributed virtual port (DVP).
<code>--add-pg-uplink <adapter_name></code> <code>--pg <port_group> <vswitch_name></code> <code>-M <adapter name></code> <code>-p <port_group> <vswitch_name></code>	Adds an uplink adapter (physical network adapter) to a port group. This command fails silently if the uplink adapter does not exist.
<code>--check <vswitch_name></code> <code>-c <vswitch_name></code>	Checks whether a virtual switch exists. Prints 1 if the switch exists and prints 0 otherwise. Use the virtual switch name, e.g. <code>vSwitch0</code> or <code>vSwitch1</code> , to specify the virtual switch.
<code>--check-pg <port_group> <vswitch_name></code> <code>-C <port_group> <vswitch_name></code>	Checks whether the specified port group exists.
<code>--delete <vswitch_name></code> <code>-d <vswitch_name></code>	Deletes a virtual switch. Running the command with this option fails if any ports on the virtual switch are still in use by VMkernel networks, <code>vswifs</code> , or virtual machines.
<code>--del-pg <port_group> <vswitch_name></code> <code>-D <port_group> <vswitch_name></code>	Deletes a port group from the virtual switch. Running the command with this option fails if the port group is in use, for example, by a virtual machine or a VMkernel network.
<code>--del-dvp-uplink <adapter></code> <code>--dvp <DVPort_id> <dvs switch_name></code> <code>-Q <adapter_name></code> <code>-V <DVPort id> <dvs switch name></code>	Deletes an uplink adapter (physical network adapter) from a port on a DVS (distributed virtual switch), also called vNetwork Distributed Switch (VDS).
<code>--del-pg-uplink <portgroup></code> <code>--pg <port_group> <vswitch_name></code> <code>-N <adapter_name></code> <code>-p <port_group> <vswitch_name></code>	Deletes an uplink adapter from a port group.
<code>--dvp</code> <code>-V</code>	Name of a distributed virtual port.
<code>--link <pnic></code> <code>-L <pnic></code>	Adds an uplink adapter (physical NIC) to a virtual switch. Running the command with this option attaches a new unused physical network adapter to a virtual switch.
<code>--list</code> <code>-l</code>	Lists all virtual switches and their port groups.
<code>--mtu <vswitch_name></code> <code>-m <vswitch_name></code>	Sets the MTU (maximum transmission unit) of the virtual switch. This option affects all physical NICs assigned to the virtual switch.
<code>--pg <port_group></code> <code>-p <port_group></code>	Provides the name of the port group for the <code>--vlan</code> option. Specify ALL to set VLAN IDs on all port groups of a virtual switch.
<code>--unlink <pnic></code> <code>-U <pnic></code>	Removes a physical NIC from a virtual switch. A physical NIC is a physical Ethernet adapter to which the virtual switch is connected. If you remove the last NIC, you lose physical network connectivity for that switch.

Table 2-13. Options for vicfg-vswitch (Continued)

Option	Description
--vihost <host> -h <host>	When you run a vSphere CLI command with the --server option pointing to a vCenter Server system, use --vihost to specify the ESX/ESXi host to run the command against.
--vlan <port_group> -v <port_group>	Sets the VLAN ID for a specific port group of a virtual switch. Setting the option to 0 disables the VLAN for this port group. If you specify this option, you must also specify the --pg option.

Examples

The following examples for vicfg-vswitch require connection options. See [“Specifying Authentication Information”](#) on page 13.

vicfg-vswitch.pl <conn_options> --add vSwitch1

Adds vSwitch1 as a virtual switch.

vicfg-vswitch.pl <conn_options> --add-pg group1 vSwitch1

Adds a port group to vSwitch1.

vicfg-vswitch.pl <conn_options> -c vSwitch1

Checks whether vSwitch1 exists. Prints 1 if the switch exists, 0 if the switch does not exist.

vicfg-vswitch.pl <conn_options> -m 9000 vSwitch1

Sets the MTU of the virtual switch vSwitch1 to 9000.

vicfg-vswitch -P vmnic1 -V 11 dvs-functional-Pos001

Adds an uplink adapter (vmnic1) to the DVS 11 at the dvs-functional-Pos001.

vicfg-vswitch.pl <conn_options> -l

Prints information about the virtual switch. See [“Sample Output from vicfg-vswitch -l.”](#)

Sample Output from vicfg-vswitch -l

The following output might result from a call to vicfg-vswitch -l.

Switch Name	Num Ports	Used Ports	MTU	Uplinks
vSwitch0	64	4	1500	vmnic2,vmnic0

PortGroup Name	VLAN ID	Used Ports	Uplinks
VM Network	0	0	vmnic0
VMkernel2	0	1	vmnic0
Management Network	0	1	vmnic0

Switch Name	Num Ports	Used Ports	MTU	Uplinks
vSwitch1	64	3	1500	vmnic1

PortGroup Name	VLAN ID	Used Ports	Uplinks
bldg1	0	1	vmkn1c1
bldg2	0	0	

System Management Commands

vSphere CLI system management commands perform system configuration.

Table 2-14. System Management Commands

Command	See
resxtp	“Using resxtp for Performance Monitoring” on page 39 and the <i>Resource Management Guide</i> .
vicfg-cfgbackup	“Configuration Information Backup with vicfg-cfgbackup” on page 39
vicfg-snmp	“SNMP Management with vicfg-snmp” on page 40
vicfg-syslog	“Syslog Server Specification with vicfg-syslog” on page 41
vihostupdate	“Update Management with vihostupdate” on page 42

Using resxtp for Performance Monitoring

The `resxtp` vSphere CLI command allows you to examine how ESX/ESXi systems use resources. You can use the command in interactive mode (default) or in batch mode. The *Resource Management Guide* explains how to use `resxtp` and provides information about available commands and display statistics.

IMPORTANT `resxtp` is supported only on Linux.

Configuration Information Backup with vicfg-cfgbackup

The `vicfg-cfgbackup` command backs up ESXi configuration data and restores them later.

Back up your host configuration before you change the configuration or upgrade the ESXi image.

The *Upgrade Guide* discusses backing up and restoring the ESXi configuration.

IMPORTANT This command is available only for ESXi hosts. The command is not available for ESX hosts and is not available through a vCenter Server system connection.

Options

[Table 2-15](#) lists the command-specific options for `vicfg-cfgbackup`. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-15. Options for `vicfg-cfgbackup`

Option	Description
<code>--force</code> <code>-f</code>	Forces the restoration of the configuration.
<code>--load <backupfile></code> <code>-l <backupfile></code>	Restores configuration from <backupfile> onto the host.
<code>--quiet</code> <code>-q</code>	Does not prompt for user confirmation.
<code>--reset</code> <code>-r</code>	Resets the host, that is, restores to factory settings.
<code>--save <backupfile></code> <code>-s <backupfile></code>	Backs up the host configuration to <backupfile>. Include the number of the build that is running on the host that you are backing up in the backup filename. If you are running the vSphere CLI from vMA, the backup file is saved locally on vMA. Local storage for backup files is safe because vMA is stored in the <code>/vmfs/volumes/<datastore></code> directory, which is separate from the ESXi image and configuration files.

Examples

The following examples for `vicfg-cfgbackup` require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-cfgbackup.pl <conn_options> -s C:\backup.txt
```

Backs up the host configuration.

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

Restores the host to factory settings.

```
vicfg-cfgbackup.pl <conn_options> -l C:\backup.txt
```

Restores the host to a configuration you previously saved to a file.

```
vicfg-cfgbackup.pl <conn_options> -l C:\backup.txt -q
```

Restores the host to a configuration you previously saved to a file, and does not prompt for confirmation.

SNMP Management with vicfg-snmp

Simple Network Management Protocol (SNMP) allows management programs to monitor and control networked devices. The *Basic System Administration* manual discusses using SNMP in your vSphere environment in some detail.

Setting up an SNMP agent for polling and notification requires that you make the following calls:

- 1 A call to `vicfg-snmp --communities` to specify at least one community.
- 2 A call to `vicfg-snmp --targets` to configure a trap destination.
- 3 A call to `vicfg-snmp --enable` to enable the SNMP service.

Setting up an SNMP agent for polling and notification requires that you make the following calls:

- 1 A call to `vicfg-snmp --communities` to specify at least one community.
- 2 A call to `vicfg-snmp --enable` to enable the SNMP service.

When adding or changing the notification configuration, run `vicfg-snmp --test` to validate the configuration.

Options

[Table 2-15](#) lists the command-specific options for `vicfg-snmp`. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-16. vicfg-snmp Command Options

Option	Description
<code>--communities <comm1>[,...]</code> <code>-c <comm1>[,...]</code>	Specifies communities, separated by commas. The settings specified using this option overwrite any previous settings.
<code>--disable</code> <code>-D</code>	Stops the SNMP service.
<code>--enable</code> <code>-E</code>	Starts the SNMP service.
<code>--port <port-number></code> <code>-p <port-number></code>	Sets the port used by the SNMP agent. The default is UDP 161. This is the port that the SNMP service uses to listen on for polling requests, such as GET requests. You can also configure the port that the SNMP agent sends data to on the target system using the <code>--targets</code> option. That port is UDP 162 by default.
<code>--reset</code> <code>-r</code>	Clears all previously-specified communities and targets.
<code>--show</code> <code>-s</code>	Displays the current SNMP configuration.

Table 2-16. vicfg-snmp Command Options (Continued)

Option	Description
--targets <hostname[@port]> </community>[,...] -t <hostname[@port]> </community>[,...]	Sets the destination for (notifications) traps. The settings specified using this flag overwrite any previous settings.
--test -T	Sends a test notification that can be used to validate the SNMP configuration.
--vihost -h	When you run a vSphere CLI command with the --server option pointing to a vCenter Server system, use --vihost to specify the ESX/ESXi host to run the command against.

Examples

The following examples for `vicfg-snmp` require connection options. See [“vSphere CLI Connection Options”](#) on page 17.

```
vicfg-snmp.pl <conn_options> -c public
```

Sets the community to public.

```
vicfg-snmp.pl <conn_options> -c my_comm1,my_comm2
```

Sets `my_comm1` and `my_comm2` as the communities, overwriting any existing communities.

```
vicfg-snmp.pl <conn_options> -E
```

Enables the SNMP service.

```
vicfg-snmp.pl <conn_options> -p 163
```

Sets the SNMP agent port to port 163.

```
vicfg-snmp.pl <conn_options> -s
```

Displays the SNMP agent configuration. Returns information like the following:

```
Current SNMP agent settings:
Enabled: 0
UDP port: 161
Communities:
Notification targets:
```

The following example shows how the commands are run in sequence:

```
vicfg-snmp -c public -t example.com@162/private --enable
# next validate your config by doing these things:
vicfg-snmp -test
walk -v1 -c public esx-host
```

Syslog Server Specification with vicfg-syslog

The `vicfg-syslog` command specifies a remote syslog server for an ESXi host.

You cannot run this command against an ESX host. For ESX hosts, users edit the syslog configuration file to customize the syslog server. The *Basic System Administration* manual discusses system logs in more detail and explains how to set them up using the vSphere Client.

Options

Table 2-17 lists the command-specific options for `vicfg-syslog`. You must also specify connection options. See “[vSphere CLI Connection Options](#)” on page 17.

Table 2-17. Options for `vicfg-syslog`

Option	Description
<code>--setport</code> <code>-p</code>	Sets the port for the syslog server. Can be used with <code>-s</code> .
<code>--setserver <sys_server></code> <code>-s <sys_server></code>	Host name or IP address of the remote syslog server. Can be used with <code>-p</code> .
<code>--show</code> <code>-i</code>	Displays the remote syslog server, if a syslog server has been set up.

Examples

The following examples for `vicfg-syslog` require connection options. See “[Specifying Authentication Information](#)” on page 13.

`vicfg-syslog.pl <conn_options> -i`

Displays the syslog server configuration.

`vicfg-syslog.pl <conn_options> -s mysyslogserver`

Makes `mysyslogserver` the syslog server for the server specified in `<conn_options>`.

`vicfg-syslog.pl <conn_options> -p <port>`

Sets the port number used by the syslog server.

Update Management with `vihostupdate`

The `vihostupdate` command applies software updates to ESX/ESXi images and installs and updates ESX/ESXi extensions such as VMkernel modules, drivers, and CIM providers.

IMPORTANT Run `vihostupdate` against ESX/ESXi 4.0 hosts. Run `vihostupdate35` against ESX/ESXi 3.5 hosts. See “[Performing Maintenance on VMware Infrastructure 3.5 Systems with vihostupdate35](#)” on page 101.

You cannot run `vihostupdate` against vCenter Server systems.

The `vihostupdate` command works with bulletins. Each bulletin consists of one or more vSphere bundles and addresses one or more issues. A bulletin is considered to be included in another bulletin if every vSphere bundle in the first bulletin meets one of these criteria:

- The vSphere bundle is included in the second bulletin.
- The vSphere bundle is obsoleted by another bundle in the second bulletin.

Towards the end of a release, bulletins include a large number of other bulletins.

Bulletins are available in bundles and in depots with associated `metadata.zip` files.

- If you use bundles, all patches and corresponding metadata are available as one ZIP file.
- If you use depots, the `metadata.zip` file points to metadata. The metadata describes the location of the files.

The command supports querying software installed on a host, listing software in a patch, scanning for bulletins that apply to a host, and installing all or selective bulletins in the patch. You can specify a patch by using a bundle ZIP file or the metadata ZIP file of a depot. The depot can be on the remote server, or you can download a bundle ZIP file and use a local depot.

`vihostupdate` supports `https://`, `http://`, and `ftp://` downloads. You can specify the protocols in the download URL for the bundle or metadata file.

See the *ESXi Upgrade Guide* for some additional information. For more information about Installing, Removing, and Updating 3rd-Party Extensions in vSphere 4.0, see the *ESXi Installable and vCenter Server Setup Guide*. An example is in the [Examples](#) section below.

IMPORTANT You can specify a comma-separated list of bulletins using the `--bundle` options. You might need to do so to install both VMware bulletins and bulletins provided by a VMware partner.

Options

[Table 2-18](#) lists the command-specific options for `vihostupdate`. You must also specify connection options. See “[vSphere CLI Connection Options](#)” on page 17.

Table 2-18. Options for `vihostupdate`

Option	Description
<code>--bulletin [<bulletin>]</code> <code>-B [<bulletin>]</code>	Bulletins to install. Use a comma-separated list. If this option is not specified, <code>vihostupdate</code> installs all bulletins.
<code>--bundle <location></code> <code>-b <location></code>	Location of the offline bundle. You can specify either <code>-b</code> or <code>-m</code> but not both. You can specify this parameter only once. Specify a comma-separated list of bulletins if instructed to do so by VMware.
<code>--install</code> <code>[<bundle> <metadata>]</code> <code>-i [<bundle> <metadata>]</code>	Installs selective bulletins from the bundle or the depot on the host. Requires either <code>-b</code> or <code>-m</code> but not both.
<code>--list [<bundle> <metadata>]</code> <code>-l [<bundle> <metadata>]</code>	Lists the bulletins in the specified bundle or depot. Requires either <code>-b</code> or <code>-m</code> but not both.
<code>--metadata</code> <code>-m</code>	Parameter to specify the location of the depot <code>metadata.zip</code> file. You can specify either <code>-b</code> or <code>-m</code> but not both. You can specify this parameter only once.
<code>--query</code> <code>-q</code>	Displays all bulletins that are already installed on the host.
<code>--remove</code> <code>-r</code>	Removes the specified bulletin from the host. Use this option for removing bulletins that are third-party or VMware extensions. Do not remove bulletins that are VMware patches or updates.
<code>--scan [<bundle> <metadata>]</code> <code>-s [<bundle> <metadata>]</code>	Scans the host for the bundle or the depot for applicable bulletins. Requires either <code>-b</code> or <code>-m</code> but not both.

Examples

You can use `vihostupdate` with bundles or with depots. The following examples require connection options. See [Table 1-2, “vSphere CLI Connection Options,”](#) on page 17.

You can update an ESX/ESXi host using bundles by running the following commands in sequence:

- 1 Find out which bulletins are installed on your ESX/ESXi host.
`vihostupdate.pl <conn_options> --query`
- 2 Find out which bulletins are available in the bundle.
`vihostupdate.pl <conn_options> --list --bundle http://<webserver>/rollup.zip`
- 3 Find out which bulletins in the bundle are applicable to your ESX/ESXi host.
`vihostupdate.pl <conn_options> --scan --bundle http://<webserver>/rollup.zip`

- 4 Install all or some bulletins from the bundle on the ESX/ESXi host. The following example installs both VMware bulletins and bulletins made available by a partner.

```
vihostupdate.pl <conn_options> --install  
    --bundle http://<server>/rollup.zip,http://<server>/rollupPartner1.zip
```

- 5 If necessary, you can remove individual bulletins.

Use this option only for removing bulletins that are third-party or VMware extensions. Do not remove bulletins that are VMware patches or updates.

```
vihostupdate.pl <conn_options> --remove --bulletin bulletin1
```

You can update your ESX/ESXi host using depots by running the following commands in sequence:

- 1 List all bulletins in the depot given the metadata.zip file location.

```
vihostupdate.pl --list --metadata http://<webserver>/depot/metadata.zip
```

- 2 Scan the depot for bulletins that are applicable to the host.

```
vihostupdate.pl --scan --metadata http://<webserver>/depot/metadata.zip
```

- 3 Install bulletins in the depot on the host:

To install all bulletins, call:

```
vihostupdate.pl --install --metadata http://<webserver>/depot/metadata.zip
```

To install selected bulletins in the specified depot on the host, use a comma-separated list. Spaces after the comment are not supported.

```
vihostupdate.pl --install --metadata http://<webserver>/depot/metadata.zip  
    --bulletin bulletin1,bulletin3
```

You can deploy a third-party bundle that you have downloaded on your web server, for example:

```
vihostupdate.pl <conn_options> --install --bundle https://<3rdParty_webserver>/Cisco_Swordfish.zip
```

Virtual Machine Management Commands

A number of vSphere CLI commands perform virtual machine management.

Table 2-19. Virtual Machine Management Commands

Command	See
svmotion	“Virtual Machine Migration with svmotion” on page 44.
vmware-cmd	Chapter 5, “Virtual Machine Management with vmware-cmd,” on page 79.

Virtual Machine Migration with svmotion

The `svmotion` command moves a virtual machine’s configuration file, and, optionally, its disks, while the virtual machine is running. *Basic System Administration* discusses how to use storage vMotion™. You can use `svmotion` to initiate migrations for virtual machines running on either ESX or ESXi hosts.

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

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. The `--disks` option relocates individual virtual disks to different datastores.

You cannot relocate a virtual disk without relocating the virtual machine configuration file.

The `svmotion` command supports both interactive or noninteractive mode.

- To use the command in interactive mode, type `svmotion --interactive`. The command prompts you for the information necessary to complete the storage migration. When you run the command in interactive mode, all other options are ignored.

IMPORTANT On Windows, use quotes around special characters.

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

```
svmotion [standard vSphere CLI 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.

Options

Table 2-20 lists the command-specific options for `svmotion`. You must also specify connection options. See “vSphere CLI Connection Options” on page 17. With `svmotion`, the `--server` option must point to a vCenter Server system.

Table 2-20. Options for `svmotion`

Option	Description
<code>--datacenter</code> <code><datacenter_name></code>	Datacenter that contains the virtual machine to be migrated. Surround the name in quotes if it contains white spaces or special characters.
<code>--disks</code> <code><virtual_disk_datastore_path></code> <code>:<new_datastore>...</code>	Locations of individual disks. The format is datastore path of the disk, colon, name of the destination datastore. If the path contains spaces or special characters, you must quote it. You can specify multiple datastore and destination pairs, separated by commas. If you do not specify this option, all virtual disks associated with a virtual machine are relocated to the same datastore as the virtual machine configuration file. Specify this option to locate individual virtual disks to different datastores. To keep a virtual disk on its current datastore, use the <code>--disks</code> option for that disk, with its current datastore as the <code><new datastore></code> . See “To determine the path to a virtual disk file” on page 46.
<code>--interactive</code>	Starts interactive mode, where <code>svmotion</code> prompts you for all arguments. In interactive mode, <code>svmotion</code> does a sanity test on each option.
<code>--vm</code> <code><VM_config_ds_path>:<new_ds></code>	Specifies which virtual machine to move and to which datastore. <code><VM config datastore path></code> is the path to the virtual machine configuration file. If the path contains spaces or other special characters, you must quote it. See “To determine the path to the virtual machine configuration file” on page 45. <code><new datastore></code> is the name of the new datastore for the virtual machine configuration file or disk.

You must specify the datastore path to the virtual machine configuration file in the `<VM config datastore path>` argument.

To determine the path to the virtual machine configuration file

- 1 In the vSphere Client inventory, select the virtual machine and click the **Summary** tab.
- 2 Click **Edit Settings** to display the Virtual Machine Properties dialog box.
- 3 Click the **Options** tab, and select **General Options**.

The Virtual Machine Configuration File text box displays the path to the virtual machine configuration file.

You must specify the virtual disk datastore path in the `<virtual disk datastore path>` option.

To determine the path to a virtual disk file

- 1 In the vSphere Client inventory, select the virtual machine to which the virtual disk belongs, and click the **Summary** tab.
- 2 Click **Edit Settings** to display the Virtual Machine Properties dialog box.
- 3 Click the **Hardware** tab and select the virtual disk from the list of devices.

The Disk File text box displays the path to the virtual disk file.

Examples

The following examples for `svmotion` require connection options. See [Table 1-2, “vSphere CLI Connection Options,”](#) on page 17. Line breaks in the examples are included for legibility only. Square brackets in the examples indicate a datastore, not an optional element.

svmotion.pl <conn_options> --interactive

Starts the interactive version of the `svmotion` command.

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

Relocates a virtual machine's storage (including disks) to `new_datastore`.

```
svmotion.pl
<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"
```

Relocates a virtual machine's configuration file to `new_datastore`, but leave the two disks (`myvm_1.vmdk` and `myvm_2.vmdk`) in `old_datastore`. The example is for Windows. For Linux, use single quotes instead of double quotes.

```
svmotion.pl <conn_options>
--datacenter RcliTstDc
--vm="[storage1] testvm/testvm.vmx:storage2"
```

Migrates the virtual machine `testvm` from `storage1` to `storage2`.

Miscellaneous Management Commands

The vSphere CLI includes commands for user management and VMkernel module manipulation, as well as the `vicfg-advcfg` command. `vicfg-advcfg` can enable or disable CIM providers, and might be used in some other situations.

Table 2-21. Miscellaneous Management Commands

Command	See
<code>vicfg-advcfg</code>	“Advanced Management with vicfg-advcfg” on page 46
<code>vicfg-module</code>	“VMkernel Module Manipulation with vicfg-module” on page 47
<code>vicfg-user</code>	“User and Group Management with vicfg-user” on page 48

Advanced Management with vicfg-advcfg

The `vicfg-advcfg` command offers a number of low-level advanced options. Most options are not intended for customer use. You might use this command when VMware Technical Support or a VMware Knowledge Base article instruct you to do so.

You can use the `vicfg-advcfg -s` option to enable and disable CIM providers.

Options

You can enable or disable CIM providers using the following option and value combinations:

Table 2-22. vicfg-advcfg Options for enabling CIM providers

Option	Description
-s 0 1 UserVars.CIMEnabled	Enables or disables all CIM providers.
-s 0 1 UserVars.CIMOEMProvidersEnabled	Enables or disables OEM CIM providers.
-s 0 1 UserVars.CIMCustomProvidersEnabled	Enables or disables IHV CIM providers.

Examples

```
vicfg-advcfg.pl <conn_options> -s 1 UserVars.CIMEnabled
```

Enables all CIM providers.

```
vicfg-advcfg.pl <conn_options> -s 1 UserVars.CIMOEMProvidersEnabled
```

Enables OEM CIM providers.

```
vicfg-advcfg.pl <conn_options> -s 0 UserVars.CIMCustomProvidersEnabled
```

Disables IHV CIM providers.

VMkernel Module Manipulation with vicfg-module

The `vicfg-module` command supports setting and retrieving VMkernel module options. This command is a vSphere CLI implementation of the `esxcfg-module` service console command but supports only some of the options `esxcfg-module` supports. The command is commonly used when VMware Technical Support, a Knowledge Base article, or VMware documentation instruct you to do so.

Options

[Table 2-23](#) lists the command-specific options for `vicfg-module`. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-23. Options for vicfg-module

Option	Description
--get-options <module_name> -g <module_name>	Returns the option string configured to be passed to the module when the module is loaded. This string is not necessarily the option string currently in use by the module.
--list -l	Lists the set of modules on the host.
--set-options <options> <module_name> -s <options> <module_name>	Specifies the option string to be passed to the module when the module is loaded.
--vihost <host> -h <host>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.

Examples

The following examples illustrate NetQueue configuration. The *ESX Configuration Guide* and the *ESXi Configuration Guide* explain NetQueue configuration in more detail. The following examples require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-module.pl <conn_options> -s "intr_type_2 rx_ring_num=8" s2io
```

Configures a supported NIC to use NetQueue.

```
vicfg-module.pl <conn_options> -g s2io
```

Verifies that the NetQueue module has been configured.

User and Group Management with vicfg-user

An ESX/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. The `vicfg-user` command supports creating, modifying, deleting, and listing local direct access users and groups of users on an ESX/ESXi host. You cannot run this command against a vCenter Server system.

User management is discussed in detail in the *ESX Configuration Guide*, the *ESXi Configuration Guide*, and the *Basic System Administration* document.

The syntax of this command differs from the syntax of other vSphere CLI commands. You specify operations as follows:

```
vicfg-user <conn_options> -e <user | group> -o <add | modify | delete | list>
```

IMPORTANT If you create a user without specifying the role (`--role`), the user has no permissions.

Options

[Table 2-24](#) lists the command-specific options for `vicfg-user`. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-24. Options for `vicfg-user`

Option	Description
<code>--addgroup <group_list></code> <code>-g <group_list></code>	Adds the user to a comma-separated list of groups.
<code>--adduser <user_list></code> <code>-u <user_list></code>	Adds the specified users to a specified group. Takes a comma-separated list of users.
<code>--entity <group user></code> <code>-e <group user></code>	Entity to perform the operation on. Specify either <code>user</code> or <code>group</code> .
<code>--group <name></code> <code>-d <name></code>	Group name of the group.
<code>--groupid <group_id></code> <code>-D <group_id></code>	Group ID of the group.
<code>--login <login_id></code> <code>-l <login_id></code>	Login ID of the user.
<code>--newpassword <p_wd></code> <code>-p <p_wd></code>	Password for the target user.
<code>--newuserid <UUID></code> <code>-i <UUID></code>	New UUID for the target user.
<code>--newusername <name></code> <code>-n <name></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>--removegroup <group_list></code> <code>-G <group_list></code>	Comma-separated list of groups to remove the target user from.
<code>--removeuser <user_list></code> <code>-U <user_list></code>	Comma-separated list of users to be removed from the target group.

Table 2-24. Options for vicfg-user (Continued)

Option	Description
<code>--role <administrator read-only no-access></code> <code>-r <administrator read-only no-access></code>	Role for the target user or group. Specify one of <code>administrator</code> , <code>read-only</code> , or <code>no-access</code> . If you create a user without assigning permissions, the user has 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 supported only against ESX. The option is not supported against ESXi.

Examples

The following examples for `vicfg-user` require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-user.pl <conn_options> -e user -o add -l user27 -p 27_password
```

Adds the user with login ID user27.

```
vicfg-user.pl <conn_options> -e user -o modify -l user27 -p 27_password -i <new user id>
-n <new user name>
```

Modifies the password, user ID, and user name for the user with login ID user27.

```
vicfg-user.pl <conn_options> -e user -o modify -l user27 -g test
```

Adds the user with login ID user27 to a group test.

```
vicfg-user.pl <conn_options> -e user -o delete -l user27
```

Removes the user with login ID user27.

```
vicfg-user.pl <conn_options> -e group -o add -d group42 -D <group id>
```

Adds group42 as a group.

```
vicfg-user.pl <conn_options> -e group -o modify -d group42 -u test
```

Adds a user test to the group group42.

```
vicfg-user.pl <conn_options> -e group -o delete -d group42
```

Removes the group group42.

```
vicfg-user.pl <conn_options> --entity group --operation add --group group42 --groupid 501
--role read-only
```

Adds group group42, with group ID 501 and role read-only.

Volume Management with vicfg-volume

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.

The *ESX Configuration Guide* and the *ESXi Configuration Guide* discuss volume resignaturing in detail.

Options

Table 2-25 lists command-specific options for `vicfg-volume`. You must also specify connection options. See “vSphere CLI Connection Options” on page 17.

Table 2-25. Options for `vicfg-volume`

Option	Description
<code>--list</code> <code>-l</code>	Lists all volumes that have been detected as snapshots or replicas.
<code>--persistent-mount <VMFS-UUID label></code> <code>-M <VMFS-UUID label></code>	Mounts a snapshot/replica volume persistently if its original copy is not online.
<code>--resignature <VMFS-UUID label></code> <code>-r <VMFS-UUID label></code>	Resignatures a snapshot/replica volume.
<code>--umount <VMFS-UUID label></code> <code>-u <VMFS-UUID label></code>	Unmounts a snapshot/replica volume.
<code>--vihost <host></code> <code>-h <host></code>	When you run a vSphere CLI command with the <code>--server</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESX/ESXi host to run the command against.

Examples

```
vicfg-volume.pl <conn_options> -l
```

Lists all volumes that have been detected as snapshots or replicas.

```
vicfg-volume.pl -M my_sample_vol
```

Mounts a snapshot/replica volume persistently.

```
vicfg-volume.pl -r my_sample_vol
```

Resignatures a snapshot/replica volume.

```
vicfg-volume.pl -u 48c826a3-12815d67-0ac6-0030485cd343
```

Unmounts a snapshot/replica volume.

File System Manipulation with `vifs`

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

IMPORTANT While there are some similarities between `vifs` and DOS or Unix file system management utilities, there are also many differences. For example, `vifs` does not support wildcard characters or current directories and, as a result, relative path names. Use `vifs` only as documented.

Instead of using this command, you can browse datastore contents and host files using a Web browser.

Connect to the following location:

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

You can view datacenter and datastore directories from this root URL.

Files and directories can be classified into three groups. The `vifs` command supports all three groups, but different operations are available for each group.

Group	Description
Host	Host configuration files. You must specify the file's unique name identifier. You specify host locations using the <code>host/<path></code> syntax.
Temp	The <code>/tmp</code> directory and files inside that directory. You specify temp locations using the <code>tmp/dir/subdir</code> syntax.
Datstores	<p>Datstore files and directories. You have two choices for specifying a datastore:</p> <ul style="list-style-type: none"> ■ Datastore prefix style: <code>'[ds_name] relative_path'</code>. For example: <code>'[myStorage1] testvms/VM1/VM1.vmx'</code> (Linux) or <code>"[myStorage1] testvms\VM1\VM1.vmx"</code> (Windows) ■ URL style: <code>/folder/dir/subdir/file?dsName=<name></code>. For example: <code>'/folder/testvms/VM1/VM1.vmx?dsName=myStorage1'</code> (Linux) or <code>"/folder/testvms/VM1/VM1.vmx?dsName=myStorage1"</code> (Windows) <p>The two example paths refer to the same virtual machine configuration file for the virtual machine VM1 inside the <code>testvms/VM1</code> directory of the <code>myStorage1</code> datastore.</p>

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

When you run `vifs`, you can specify the operation name and argument and also one of the standard connection options discussed in [Table 1-2](#). 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 by `vifs`.

Options

[Table 2-26](#) lists the command-specific options for `vifs`. 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. See [“vSphere CLI Connection Options”](#) on page 17.

Table 2-26. Options for `vifs`

Command	Description	For...	Syntax
<code>--copy</code> <code>-c <source></code> <code><target></code>	Copies a file in a datastore to another location in a datastore. The <code><source></code> must be a remote source path, the <code><target></code> a remote target path or directory. The <code>--force</code> option replaces existing destination files.	Datstore 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 <remote_dir></code>	Lists the contents of a datastore directory.	Datstore 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> .	Datstore Temp	<code>copy src_file_path</code> <code>dst_file_path [--force]</code>
<code>--get</code> <code>-g <remote_path></code> <code><local_path></code>	Downloads a file from the ESX/ESXi host to the machine on which you run the vSphere CLI. This operation uses HTTP GET.	Datstore Host	<code>get src_dstore_file_path</code> <code>dst_local_file_path</code> <code>get src_d_store_dir_path</code> <code>dst_local_file_path</code>
<code>--listdc</code> <code>-C</code>	Lists the datacenter paths available on an ESX/ESXi system.	Datstore Host	
<code>--listds</code> <code>-S</code>	Lists the datastore names on the ESX/ESXi system. When multiple data centers are available, you can use the <code>--dc (-Z)</code> argument to specify the name of the datacenter from which you want to list the datastore.	Datstore Host	<code>vifs --listds</code>
<code>--mkdir</code> <code>-M <remote_dir></code>	Creates a directory in a datastore. This operation fails if the parent directory of <code>dst_datastore_file_path</code> does not exist.	Datstore Temp	<code>mkdir dst_directory_path</code>

Table 2-26. Options for vifs (Continued)

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

Examples

Running `vifs` works only when you are connected directly to an ESX/ESXi host. The command does not work when you connect to a vCenter Server system and attempt to connect to the ESX/ESXi host through the vCenter Server system. On Linux, use single quotes for directory names. On Windows, use double quotes.

`vifs` examples require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vifs.pl <conn_options> --copy '[myvmfs] dir_1/my_text' '[myvmfs] dir_3/my_text'
```

Copies the `my_text` file from `dir_1` to `dir_3`.

```
vifs.pl <conn_options> --dir '[myvmfs] dir_3'
```

Lists the contents of directory `dir_3`.

```
vifs.pl <conn_options> --copy '[myvmfs] dir_1/my_text' '[myvmfs] dir_3/my_text' --force
```

Copies the `my_text` file into `dir_3`. If you use this command, and a file named `my_text` already exists in `dir_3`, the command overwrites the existing file because of the `--force` option.

```
vifs.pl <conn_options> --mkdir '[myvmfs] new_dir'
```

Creates the directory called `new_dir`.

```
vifs.pl <conn_options> --put /root/test_put '[myvmfs] new_dir/test_put'
```

Places a copy of the local `test_put` file into the `new_dir` directory of the specified ESX/ESXi host.

```
vifs.pl <conn_options> --rm '[myvmfs] new_dir/test_put'
```

Removes the `test_put` file from the `new_dir` folder.

```
vifs.pl <conn_options> --rmdir '[myvmfs] new_dir'
```

Removes the `new_dir` folder.

```
vifs.pl <conn_options> --get '[myvmfs] dir_1/my_text' /root/my_text
```

Retrieves the file `my_text` from the ESX/ESXi host and places it in the local machine's root folder.

```
vifs.pl <conn_options> --move '[myvmfs] dir_1/my_text' '[myvmfs] dir_3/my_text'
```

Moves the `my_text` file from `dir_1` to `dir_3`.

```
vifs.pl <conn_options> --listds
```

Lists the names of all datastores on the server specified in the configuration file. You can use each name that has been returned to refer to datastore paths using square bracket notation, as follows:

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

File System Management with vmkfstools

3

You use the `vmkfstools` vSphere CLI to create and manipulate virtual disks, file systems, logical volumes, and physical storage devices on an ESX/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.

IMPORTANT The `vmkfstools` vSphere CLI supports most but not all the options supported by the `vmkfstools` service console command. See VMware Knowledge Base article 1008194.

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

You can run the `vmkfstools` vSphere CLI as you would other vSphere CLI commands. Specify the ESX/ESXi host to run the command against and specify additional connection options as discussed in [“Specifying Authentication Information”](#) on page 13.

In most cases, you do not have to log in as the root user to run the `vmkfstools` command. However, some commands, such as the file system commands, might require root user login.

You can perform most `vmkfstools` operations using the vSphere Client instead of the command line. See the *ESX Configuration Guide* and the *ESXi Configuration Guide*.

This chapter includes the following topics:

- [“vmkfstools Command Syntax”](#) on page 53
- [“vmkfstools File System Options”](#) on page 55
- [“vmkfstools Virtual Disk Options”](#) on page 57

vmkfstools Command Syntax

The `vmkfstools` command supports specifying one or more command-line options, associated arguments, and the target.

```
vmkfstools <conn_options> <options> <target>
```

Option	Description	See
<conn_options>	Connection parameters.	“vSphere CLI Connection Options” on page 17.
<options>	One or more command-line options and associated values.	“Supported Command-Specific Options” on page 54.
<target>	Partition, device, or path to apply the command to.	“Supported Targets” on page 55.

The long and short (single letter) forms of options are equivalent. For example:

```
vmkfstools --createfs vmfs3 --blocksize 2m naa.<naa_ID>:1
vmkfstools -C vmfs3 -b 2m naa.<naa_ID>:1
```

Supported Command-Specific Options

[Table 3-1](#) lists `vmkfstools` file system options, a brief description, and a pointer to where the option is discussed.

Table 3-1. `vmkfstools` File System Option Overview

Option	Description	See
<code>--blocksize</code> <code>-b</code>	Uses the specified size for file system creation. Used with <code>--createfs</code> .	“VMFS File System Creation” on page 55.
<code>--createfs</code> <code>-C</code>	Creates a VMFS file system.	“VMFS File System Creation” on page 55.
<code>--queryfs</code> <code>-P</code>	Lists attributes of a file system.	“Attribute Listing for a VMFS Volume” on page 56.
<code>--setfsname</code> <code>-S</code>	Sets the label for the file system. Used with <code>--createfs</code> .	“VMFS File System Creation” on page 55.
<code>--extendfs</code> <code>-Z</code>	Extends the VMFS file system.	“File System Partition Extension by Spanning” on page 57.

[Table 3-2](#) lists `vmkfstools` virtual disk options, a brief description, and a pointer to where each option is discussed.

Table 3-2. `vmkfstools` Virtual Disk Option Overview

Option	Description	See
<code>--adaptertype</code> <code>-a</code>	Uses the specified type for disk creation. Used with <code>-c</code> and <code>-i</code> .	“Creating Virtual Disks” on page 58.
<code>--clonevirtualdisk</code> <code>-i</code>	Clones the specified virtual disk.	“Cloning Virtual or Raw Disks” on page 59.
<code>--createrdm</code> <code>-r</code>	Maps a raw disk to a file on a VMFS file system.	“Creating Virtual Compatibility Mode Raw Device Mappings” on page 61.
<code>--createrdmpassthru</code> <code>-z</code>	Maps a passthrough raw disk to a file on a VMFS file system.	“Creating Physical Compatibility Mode Raw Device Mappings” on page 61.
<code>--createvirtualdisk</code> <code>-c</code>	Creates a virtual disk.	“Creating Virtual Disks” on page 58.
<code>--deletevirtualdisk</code> <code>-U</code>	Deletes the specified virtual disk.	“Deleting Virtual Disks” on page 59.
<code>--diskformat</code> <code>-d</code>	Uses the specified format for disk creation. Used with <code>-c</code> and <code>-i</code> .	“Supported Disk Formats” on page 57.
<code>--extendvirtualdisk</code> <code>-X</code>	Extends the specified virtual disk.	“Extending Virtual Disks” on page 60.
<code>--geometry</code> <code>-g</code>	Displays virtual disk geometry.	“Displaying Virtual Disk Geometry” on page 62.
<code>--inflatedisk</code> <code>-j</code>	Converts a thin virtual disk to <code>eagerzeroedthick</code> format, preserving all existing data.	“Inflating Thin Virtual Disks” on page 59.
<code>--queryrdm</code> <code>-q</code>	Lists the attributes of a raw disk mapping.	“Listing RDM Attributes” on page 61.
<code>--renamevirtualdisk</code> <code>-E</code>	Renames the specified virtual disk.	“Renaming Virtual Disk” on page 59.
<code>--writezeros</code> <code>-w</code>	Cleans the virtual disk by writing zeros over all its data.	“Initializing Virtual Disks” on page 58.

Supported Targets

You can specify the target of the operation specified in <options> as a file system, partition, or virtual disk. You can use a relative or absolute path name in the /vmfs hierarchy.

File System Target

Specify a VMFS file system or file using an absolute or relative path that names a directory symbolic link, raw device mapping, or a file under /vmfs.

```
VMFS file system  /vmfs/volumes/<file_system_UUID>
                  /vmfs/volumes/<file_system_label>

VMFS file         /vmfs/volumes/<file system label|file system UUID>/[dir]/myDisk.vmdk

                  You must use an absolute path name starting with /vmfs/volumes. For example,
                  /vmfs/volumes/datastore1/rh9.vmdk
```

See [“vmkfstools File System Options”](#) on page 55.

Disk Partition Target

Specify a disk partition using `naa.<naa_ID>:P` where `naa.<naa_ID>` is the device ID returned by the storage array and `P` is an integer that represents the partition number. The partition digit must be greater than zero and must correspond to a valid VMFS partition of type `fb`.

See [“vmkfstools Virtual Disk Options”](#) on page 57.

Device Target

Specify a device or logical volume using a path name in an ESX/ESXi device file system. The name begins with /vmfs/devices, which is the mount point of the device file system. Each device type has submounts, for example:

- /vmfs/devices/disks for local or SAN-based disks.
- /vmfs/devices/lvm for ESX/ESXi logical volumes.
- /vmfs/devices/generic for generic SCSI devices, such as tape drives.

vmkfstools File System Options

Using `vmkfstools` file system commands, you can create, query, and extend a VMFS file system. The options do not apply to NFS file systems. You must also specify connection options. See [“vSphere CLI Connection Options”](#) on page 17.

The long and short (single letter) forms of options are equivalent. For example, the following commands are identical:

```
vmkfstools.pl <conn_options> --createfs vmfs3 --blocksize 2m naa.<naa_ID>:1
vmkfstools.pl <conn_options> -C vmfs3 -b 2m naa.<naa_ID>:1
```

You can perform many of file system management tasks through the vSphere Client.

VMFS File System Creation

The `-C` option creates a VMFS file system on the specified partition, such as `naa.<naa_ID>:1`. The partition becomes the file system's head partition. You can have only one VMFS volume for a LUN.

```
-C --createfs vmfs3
    -b --blocksize <block_size>kk|mM
    -S --setfsname <fsName>
```

IMPORTANT This prompt does not ask for confirmation. Check carefully before you run the command to avoid erasing important data.

VMFS-2 file systems are read-only on any ESX/ESXi host. You cannot create or modify VMFS-2 file systems but you can read files stored on VMFS-2 file systems.

[Table 3-3](#) lists the suboptions you can use with the `-C` option.

Table 3-3. Suboptions for `vmkfstools -C`

Option	Description
<code>-b --blocksize</code>	Defines the block size for the VMFS-3 file system. The default file block size is 1MB. Valid block sizes for VMFS3 are 1, 2, 4, 8MB When entering a size, indicate the unit type by adding a suffix such as m or M. The unit type is not case sensitive— <code>vmkfstools</code> interprets either m or M to mean megabytes and k or K to mean kilobytes.
<code>-S --setfsname</code>	Defines the volume label of a VMFS volume for the VMFS-3 file system you are creating. Use this suboption only with the <code>-C</code> option. The label you specify can be up to 128 characters long and cannot contain any leading or trailing blank spaces. After you have defined a volume label, you can use it whenever you specify the VMFS volume in a call to <code>vmkfstools</code> . The volume label appears in listings generated for the Linux <code>ls -l</code> command and as a symbolic link to the VMFS volume under the <code>/vmfs/volumes</code> directory. You can change the VMFS volume label using the vSphere Client UI.

Examples

The following examples require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -C vmfs3 -b 1m -S my_vmfs /vmfs/devices/disks/naa.<naa_ID>:1
```

Creates a new VMFS-3 file system named `my_vmfs`. The file block size is 1MB.

```
vmkfstools.pl <conn_options> -C vmfs3 -S my_vmfs naa.<naa_ID>:4
```

```
vmkfstools.pl <conn_options> --createfs vmfs3 --setfsname my_vmfs naa.<naa_ID>:4
```

```
vmkfstools.pl <conn_options> --createfs vmfs3 --blocksize 1m --setfsname my_vmfs naa.<naa_ID>:4
```

```
vmkfstools.pl <conn_options> --createfs vmfs3 -b 4m --setfsname my_vmfs naa.<naa_ID>:4
```

Attribute Listing for a VMFS Volume

The `-P` option lists the attributes of a file or directory on a VMFS volume.

```
-P --queryfs
```

The listed attributes include the VMFS version number (VMFS-2 or VMFS-3), the number of extents in the specified VMFS volume, the volume label if any, the UUID, and a listing of the device names where each extent resides.

Example

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> --queryfs /vmfs/volumes/my_vmfs
```

This command might return the following:

```
VMFS-3.33 file system spanning 1 partitions.
Capacity : 65229815808, 64641564672 avail
File system label : my_vmfs
UUID : 46fd1460-6ec4e2b8-e048-000e0c7f4088
Path : /vmfs/volumes/46fd1460-6ec4e2b8-e048-000e0c7f4088
Partitions spanned: naa.xxxxxxxxxxxxxxxxxxxxxx:3
```

If any device backing VMFS file system goes offline, the number of extents and the available space change accordingly.

File System Partition Extension by Spanning

The `-Z` option extends the VMFS file system with the specified head partition by spanning it across the partition specified by `<span-partition>`.

`-Z | --extendfs <span-partition>`

The operation erases existing data on the spanned partition. A VMFS file system can have at most 32 partitions.

This options does not work on VMFS-2 volumes because VMFS-2 volumes are read-only in ESX/ESXi 3.0 and later.



CAUTION When you run this option, you lose all data on the SCSI device you specified in `<span-partition>`.

Example

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -Z /vmfs/devices/disks/naa.<naa_id_1>:1
/vmfs/devices/disks/naa.<naa_id_2>:3
```

Extends the logical file system by allowing it to span to a new partition. The extended file system spans two partitions, `naa.<naa_id_1>:1` and `naa.<naa_id_2>:3`. In this example, `/vmfs/devices/disks/naa.<naa_id_2>:3` is the head partition of the existing VMFS-3 file system. `naa.<naa_id_1>:1` is the partition to be added.

vmkfstools Virtual Disk Options

Virtual disk options support set up, migration, and management of virtual disks stored in VMFS-2, VMFS-3, and NFS file systems. You can also perform most of these tasks through the vSphere Client.

Supported Disk Formats

When you create or clone a virtual disk, you can use the `-d --diskformat` suboption to specify the format for your disk. [Table 3-4](#) lists the supported formats.

Table 3-4. Supported Disk Formats

Format	Description
zeroedthick (default)	Space required for the virtual disk is allocated during creation. Any data remaining on the physical device is not erased during creation, but will be zeroed out on demand at a later time on first write from the virtual machine. The virtual machine does not read stale data from disk.
eagerzeroedthick	Space required for the virtual disk is allocated at creation time. In contrast to <code>zeroedthick</code> format, the data remaining on the physical device is zeroed out during creation. It might take much longer to create disks in this format than to create other types of disks.
thin	Thin-provisioned virtual disk. Space required for the virtual disk is not allocated during creation but is supplied, zeroed out, on demand at a later time.
rdm	Virtual compatibility mode raw disk mapping.
rdmp	Physical compatibility mode (pass-through) raw disk mapping.
2gbsparse	Sparse disk with 2GB maximum extent size. You can use disks in this format with other VMware products such as VMware Workstation. You cannot power on a sparse disk on an ESX/ESXi host unless you first reimport the disk in a compatible format, such as <code>thin</code> , with <code>vmkfstools</code> .

With `-c --createvirtualdisk`, `vmkfstools` accepts `zeroedthick`, `eagerzeroedthick`, and `thin`.

With `-i --clonevirtualdisk`, `vmkfstools` accepts `zeroedthick`, `thin`, `eagerzeroedthick`, `rdm:<device>`, `rdmp:<device>`, and `2gbsparse`.

With NFS files, `vmkfstools` supports only `thin`, `zeroedthick`, and `2gbsparse`. Because the NFS server and not the ESX/ESXi system decides the allocation policy, `zeroedthick`, and `thin` usually have the same result. The default allocation policy on most NFS servers is `thin`.

Creating Virtual Disks

The `-c` option creates a virtual disk at the specified location on a VMFS volume. You must specify the size of the virtual disk.

```
-c --createvirtualdisk <size>[kK|mM|gG]
    -a --adaptype [buslogic|lsilogic|ide] <srcfile>
    -d --diskformat [thin|zeroedthick|eagerzeroedthick]
```

When you run `vmkfstools -c`, `--adaptype` defaults to `buslogic` and `--diskformat` defaults to `zeroedthick`.

Indicate the unit type for `<size>` by adding a suffix of `k` (kilobytes), `m` (megabytes), or `g` (gigabytes). The unit type is not case sensitive—`vmkfstools` interprets either `k` or `K` to mean kilobytes. If you do not specify a unit type, `vmkfstools` defaults to bytes.

The `vmkfstools` command supports the following suboptions in conjunction with `-c`.

- `-a` specifies the device driver that is used to communicate with the virtual disks. You can select `BusLogic`, `LSI Logic SCSI`, or `IDE` drivers.
- `-d` specifies disk formats, one of `zeroedthick`, `thin`, `eagerzeroedthick`. [“Supported Disk Formats”](#) on page 57 discusses all supported disk formats, including those supported by `createvirtualdisk`.

Examples

The following examples require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -c 2g /vmfs/volumes/my_vmfs/myOS.vmdk
```

Creates a two-gigabyte virtual disk file named `myOS.vmdk` on the VMFS file system named `myVMFS`. This file represents an empty virtual disk a virtual machine can access.

```
vmkfstools.pl <conn_options> --createvirtualdisk 20m /vmfs/volumes/store1/test.vmdk
```

Creates a 20MB virtual disk named `test.vmdk`.

```
vmkfstools.pl <conn_options> --createvirtualdisk 20m -d thin -a lsilogic /vmfs/volumes/M1/test.vmdk
```

Creates a virtual disk associated with the specified adapter.

```
vmkfstools.pl <conn_options> -c 200m /vmfs/volumes/my_vmfs/test01.vmdk
```

Creates a 200MB virtual disk named `test01.vmdk` on the VMFS file system named `my_vmfs`.

Initializing Virtual Disks

The `-w` option cleans the virtual disk by writing zeros over all its data.

```
-w --writezeros
```

Depending on the size of your virtual disk and the I/O bandwidth to the device hosting the virtual disk, completing this command might take a long time.



CAUTION When you use this command, you lose any existing data on the virtual disk.

Examples

The following examples require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -w /vmfs/volumes/my_vmfs/test01.vmdk
```

```
vmkfstools.pl <conn_options> --writezeros /vmfs/volumes/my_vmfs/text02.vmdk
```

Inflating Thin Virtual Disks

The `-j` option converts a thin virtual disk to `eagerzeroedthick` format, and preserves all existing data. Any blocks that were not allocated are allocated and zeroed out.

`-j --inflatedisk`

See [“Supported Disk Formats”](#) on page 57.

Examples

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> --inflatedisk /vmfs/volumes/myvmfs/thin.vmdk
```

Deleting Virtual Disks

The `-U` option deletes files associated with the virtual disk at the specified path on the VMFS volume.

`-U --deletevirtualdisk`

Example

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -U /vmfs/volumes/store/test.vmdk
```

Deletes the virtual disk `test.vmdk`.

Renaming Virtual Disk

The `-E` option renames the virtual disk file.

`-E --renamevirtualdisk <oldName> <newName>`

You must specify the original filename or file path `<oldName>` and the new filename or file path `<newName>`.

Examples

The following examples require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -E /vmfs/volumes/myvmfs/test.vmdk /vmfs/volumes/store/renamed.vmdk
```

```
vmkfstools.pl <conn_options> -E /vmfs/volumes/myvmfs/my_OS.vmdk
    /vmfs/volumes/myvmfs/my_new_OS.vmdk
```

```
vmkfstools.pl <conn_options> --renamevirtualdisk
    /vmfs/volumes/myvmfs/my_OS.vmdk /vmfs/volumes/myvmfs/my_new_OS.vmdk
```

Cloning Virtual or Raw Disks

The `-i` option creates a copy of a virtual disk or a raw disk that you specify.

```
-i --clonevirtualdisk <srcfile> <destfile>
    -d --diskformat [zeroedthick| thin| eagerzeroedthick| rdm:<device>|rdmp:<device>| 2gbsparse]
    -a --adapertype <type>
```

The `--diskformat` option specifies the disk format for the copy. If you do not specify `--diskformat`, it defaults to `zeroedthick`. If you do not specify `--adapertype`, it defaults to `buslogic`. See [“Supported Disk Formats”](#) on page 57.

IMPORTANT To clone the redo logs of an ESXi host while preserving their hierarchy, use the `vifs -C` command instead.

Example

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools <conn_options> -i /vmfs/volumes/templates/gold-master.vmdk
/vmfs/volumes/myVMFS/myOS.vmdk -d thin -a lsilogic
```

Clones the contents of a master virtual disk from the template repository to a virtual disk file named myOS.vmdk on the file system myVMFS.

Migrating VMware Workstation and VMware GSX Server Virtual Machines

You cannot use the vSphere Client to migrate virtual machines created with VMware Workstation or VMware GSX Server to your ESX/ESXi system. However, you can use the `vmkfstools -i` command to import the virtual disk into your ESX/ESXi system. You can then attach this disk to a new virtual machine that you create in the ESX/ESXi system. You must import the virtual disk first, because you cannot power on disks exported in 2gbsparse format on an ESX/ESXi host.

Migrating VMware Workstation and GSX Server Virtual Machines requires the following tasks:

- 1 Importing a VMware Workstation or GSX Server disk into your `/vmfs/volumes/myVMFS/` directory or any subdirectory using `vmkfstools`.
- 2 Creating a new virtual machine using the **Custom** configuration option in the vSphere Client.
- 3 Selecting **Use an existing virtual disk** and attaching the VMware Workstation or GSX Server disk that you imported during disk configuration.

Extending Virtual Disks

The `-X` option extends the size of a disk allocated to a virtual machine after the virtual machine has been created.

```
-X --extendvirtualdisk [-d eagerzeroedthick] <newSize>[kK|mM|gG]
```

You must power off the virtual machine that uses this disk file before you enter this command. You might have to update the file system on the disk so that the guest operating system can recognize and use the new size of the disk and take advantage of the extra space.

IMPORTANT `newSize` defines the entire new size, not just the increment you add to the disk.

You specify the `newSize` option in kilobytes, megabytes, or gigabytes by adding a suffix of `k` (kilobytes), `m` (megabytes), or `g` (gigabytes). The unit type is not case sensitive. `vmkfstools` interprets either `k` or `K` to mean kilobytes. If you do not specify a unit type, `vmkfstools` defaults to kilobytes.

Do not extend the base disk of a virtual machine that has snapshots associated with it. If you do, you can no longer commit the snapshot or revert the base disk to its original size.

By default, any disk, regardless of format, is extended as `zeroedthick`. You can specify `-d eagerzeroedthick` to change the format to `eagerzeroedthick`. Typically, it makes sense only to extend `eagerzeroedthick` disks as `eagerzeroedthick` because these virtual disks are used for fault tolerance and clustering and have to be preallocated and zeroed out up front. `-d` allows only `eagerzeroedthick`, it does not allow other disk formats.

Examples

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -X 5g <disk name>.vmdk
```

Extends a 4GB virtual disk by 1GB.

```
vmkfstools.pl <conn_options> -X 50M /vmfs/volumes/my_newVMFS/my_disk.vmdk
```

Creating Virtual Compatibility Mode Raw Device Mappings

The `-r` option creates a Raw Device Mapping (RDM) file in virtual compatibility mode on a VMFS-3 volume and maps a raw disk to this file.

```
-r --createrdm <device>
```

After this mapping is established, you can access the raw disk as you would a normal VMFS virtual disk. The file length of the mapping is the same as the size of the raw disk it points to.

When you specify `naa.<naa_id>`, the entire raw disk is used. Use the following format:

```
/vmfs/devices/disks/naa.<naa_id>
```

For details on configuring and using RDMs, see the *ESX Configuration Guide* and the *ESXi Configuration Guide*.

All VMFS-3 file-locking mechanisms apply to RDMs.

Examples

The following examples require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -r /vmfs/devices/disks/naa.<naa_id>  
/vmfs/volumes/storage1/rdm210.vmdk
```

Creates a virtual compatibility mode RDM file `/vmfs/volumes/storage1/rdm210.vmdk` and maps the `/vmfs/devices/disks/naa.<naa_id>` raw disk to that file.

```
vmkfstools.pl <conn_options> -r /vmfs/devices/disks/naa.<naa_id> my_rdm.vmdk
```

Creates an RDM file named `my_rdm.vmdk` and maps the `naa.<naa_id>` raw disk to that file. You can configure a virtual machine to use the `my_rdm.vmdk` mapping file by adding the following lines to the virtual machine configuration file:

```
scsi0:0.present = TRUE  
scsi0:0.fileName = /vmfs/volumes/myVMFS/my_rdm.vmdk
```

Creating Physical Compatibility Mode Raw Device Mappings

The `-z` option lets you map a physical compatibility mode raw device to a file on a VMFS volume.

```
-z --createrdmpassthru <device>
```

The mapping lets a virtual machine bypass ESX/ESXi SCSI command filtering when accessing its virtual disk. This type of mapping is useful when the virtual machine needs to send proprietary SCSI commands, for example, when the virtual machine runs SAN-aware software.

After you establish this type of mapping, you can use the mapping to access the raw disk just as you would any other VMFS virtual disk. The entire raw device is used. Use the following format:

```
/vmfs/devices/disks/naa.<naa_id>
```

Example

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vmkfstools.pl <conn_options> -z /vmfs/devices/disks/naa.<naa_id>  
/vmfs/volumes/storage1/rdmpass.vmdk
```

Creates a physical compatibility mode RDM file named `rdmpass.vmdk` and maps the `naa.<naa_id>` to that file. You cannot use the name of a file that already exists.

Listing RDM Attributes

The `-q` option lists the attributes of an RDM.

```
-q --queryrdm
```

This option prints the name of the raw disk RDM. The option also prints other identification information, like the disk ID, for the raw disk.

Displaying Virtual Disk Geometry

The `-g` option gets information about the geometry of a virtual disk.

`-g --geometry`

The form is `Geometry information C/H/S`, where C is the number of cylinders, H is the number of heads, and S is the number of sectors.

IMPORTANT When importing VMware Workstation virtual disks to ESX/ESXi host, you might see a disk geometry mismatch message. Geometry mismatch might also cause problems loading a guest operating system or running a newly created virtual machine.

iSCSI Storage Management with vicfg-iscsi

4

ESX/ESXi systems include iSCSI technology to access remote storage using an IP network. 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. From the point of view of the virtual machine, the device appears as a locally attached SCSI drive.

You can manage iSCSI storage using the vSphere Client UI, as discussed in the *ESX Configuration Guide*, the *ESXi Configuration Guide* and the *iSCSI SAN Configuration Guide*, or using the `vicfg-iscsi` vSphere CLI command discussed in this chapter.

The chapter includes the following topics.

- [“vicfg-iscsi Command Syntax”](#) on page 63
- [“vicfg-iscsi Options”](#) on page 64
- [“vicfg-iscsi Discovery Target Options”](#) on page 67
- [“vicfg-iscsi Authentication Options”](#) on page 69
- [“vicfg-iscsi Information Retrieval Options”](#) on page 71
- [“vicfg-iscsi Physical Network Portal Options”](#) on page 72
- [“vicfg-iscsi Network Options”](#) on page 72
- [“vicfg-iscsi Initiator Options”](#) on page 73
- [“vicfg-iscsi iSCSI Parameter Options”](#) on page 74
- [“vicfg-iscsi Software iSCSI Options”](#) on page 75
- [“Tasks Required for Software iSCSI Setup”](#) on page 76

vicfg-iscsi Command Syntax

Commands for iSCSI management usually include an option, a suboption, an optional parameter, and the adapter name. For each option, the short and the long forms are equivalent. The commands have the following syntax:

```
vicfg-iscsi <conn-params> [option][suboption][parameters][<adapter_name>]
```

Option is one of the following:

- `-D --discovery`
- `-S --static`
- `-A --authentication`
- `-P --phba`
- `-T --target`
- `-L --lun`

- `-N --network` (Hardware iSCSI only)
- `-p --pnp` (Hardware iSCSI only)
- `-I --iscsiname`
- `-W --parameter`
- `-E --swiscsi`
- `-H --adapter`

Suboption is one of the following operations:

- `-l --list`
- `-a --add`
- `-r --remove`

Parameters differ depending on the option and suboption used.

With the exception of `--adapter` and `--help`, all commands require the `<adapter_name>` argument. The adapter name should be the name that the ESX/ESXi host assigned or configured for the software or hardware iSCSI initiator. The *ESX Configuration Guide* and the *ESXi Configuration Guide* discusses iSCSI initiators in a vSphere environment.

You can use `--list` to find the adapter name.

```
vicfg-iscsi --adapter --list
vicfg-iscsi -H -l
```

vicfg-iscsi Options

`vicfg-iscsi` supports a comprehensive set of options, listed in [Table 4-1](#). Each option is discussed in more detail later in the chapter.

Table 4-1. Options for `vicfg-iscsi`

Option	Suboptions	Description
<code>-A --authentication</code>	<pre> -c <level> -m <auth_method> -b -u <ma_username> -w <ma_password> [-i <stor_ip_addr stor_hostname> [:<portnum>] [-n <iscsi_name>]] <adapter_name> --level <level> --method <auth_method> --mutual --auth_username <ma_username> --auth_password <ma_password> [--ip <stor_ip_addr stor_hostname> [:<portnum>] [--name <iscsi_name>]] <adapter_name> </pre>	Enables mutual authentication. You must enable authentication before you can enable mutual authentication.
<code>-A --authentication</code>	<pre> -c <level> -m <auth_method> -u <auth_u_name> -w <a_password> [-i <stor_ip_addr stor_hostname> [:<portnum>] [-n <iscsi_name>]] <adapter_name> --level <level> --method <auth_method> --auth_username <auth_u_name> --auth_password <auth_password> [--ip <stor_ip_addr stor_hostname> [:<portnum>] [--name <iscsi_name>]] <adapter_name> </pre>	Enables authentication using the specified options.

Table 4-1. Options for vicfg-iscsi (Continued)

Option	Suboptions	Description
-A --authentication		
	-l <adapter_name> --list <adapter_name>	Lists supported and enabled authentication methods.
-D --discovery		
	-a -i <stor_ip_addr stor_hostname[:<portnum>] <adapter_name> --add --ip <stor_ip_addr stor_hostname> [:<portnum>] <adapter_name>	Adds a dynamic discovery address.
-D --discovery		
	-l <adapter_name> --list <adapter_name>	Lists dynamic discovery addresses.
-D --discovery		
	-r -i <stor_ip_addr stor_hostname>[:<portnum>] <adapter_name> --remove --ip <stor_ip_addr stor_hostname> [:<portnum>] <adapter_name>	Removes a dynamic discovery address.
-H		
	-l [<adapter_name>] --list [<adapter_name>]	Lists all iSCSI adapters or a specified adapter.
-L --lun		
	-l <adapter_name> --list <adapter_name>	Lists LUN information.
-L --lun		
	-l -t <target_ID> <adapter_name> --list --target_id <target_id> <adapter_name>	Lists LUN information for a specific target.
-N --network (Hardware iSCSI only)		
	-l <adapter_name> --list <adapter_name>	Lists network properties.
-N --network (Hardware iSCSI only)		
	-i <ip_addr> <adapter_name> --ip <ip_addr> <vmhba>	Sets the HBA IPv4 address to ip_addr.
-N --network (Hardware iSCSI only)		
	-s <subnet_mask> <adapter_name> --subnetmask <subnet_mask> <adapter_name>	Sets the HBA network mask to subnet_mask.
-N --network (Hardware iSCSI only)		
	-g <default_gateway> <adapter_name> --gateway <default_gateway> <adapter_name>	Sets the HBA gateway to default_gateway.
-N --network (Hardware iSCSI only)		
	-i <ip_addr> -s <subnet mask> -g <default_gateway> <adapter_name> --ip <ip_addr> --subnetmask <subnet_mask> --gateway <default_gateway> <adapter_name>	Sets the IP address, subnet mask, and default gateway in one command.
-p --pnp (Hardware iSCSI only)		
	-l <adapter_name> --list <adapter_name>	Lists physical network portal options.
-p --pnp (Hardware iSCSI only)		
	-M <mtu_size> <adapter_name> --mtu <mtu-size> <adapter_name>	Sets physical network portal options.

Table 4-1. Options for vicfg-iscsi (Continued)

Option	Suboptions	Description
-I --iscsiname		
	-a <alias_name> <adapter_name> --alias <alias_name> <adapter_name>	Sets the iSCSI initiator alias.
-I --iscsiname		
	-n <iscsi_name> <adapter_name> --name <iscsi_name> <adapter_name>	Sets the iSCSI initiator name.
-I --iscsiname		
	-l <adapter_name> --list <adapter_name>	Lists iSCSI initiator options.
-S --static		
	-l <adapter_name> --list <adapter_name>	Lists static discovery addresses.
-S --static		
	-r -i <stor_ip_addr stor_hostname> [:<portnum>] -n <target_name> <adapter_name> --remove --ip <stor_ip_addr stor_hostname> [:<portnum>] -name <target_name> <adapter_name>	Removes a static discovery address.
-S --static		
	-a -i <stor_ip_addr stor_hostname> [:<portnum>] -n <target_name> <adapter_name> --add --ip <stor_ip_addr stor_hostname> [:<portnum>] -name <target_name> <adapter_name>	Adds a static discovery address.
-P --phba		
	-l <adapter_name> --list <adapter_name>	Lists external, vendor-specific properties of an iSCSI adapter.
-T --target		
	-l <adapter_name> --list <adapter_name>	Lists target information.
-W --parameter		
	-l [-i <stor_ip_addr stor_hostname> [:<portnum>] [-n <iscsi_name>]] <adapter_name> --list [--ip <stor_ip_addr stor_hostname> [:<portnum>] [--name <iscsi_name>]] <adapter_name>	Lists iSCSI parameter information.
-W --parameter		
	-l -k [-i <stor_ip_addr stor_hostname> [:<portnum>] [-n <iscsi_name>]] <adapter_name> --list --detail [--ip <stor_ip_addr stor_hostname> [:<portnum>] [--name <iscsi_name>]] <adapter_name>	Lists iSCSI parameter details.
-W --parameter		
	-W -j <name>=<value> -i <stor_ip_addr stor_hostname> [:<port_num>] [-n <iscsi_name>]] <adapter_name> --parameter --set <name>=<value> --ip <stor_ip_addr stor_hostname> [:<port_num>] [--name <iscsi_name>]] <adapter_name>	Sets iSCSI parameters.

Table 4-1. Options for vicfg-iscsi (Continued)

Option	Suboptions	Description
-W --parameter		
	-W -o <param_name> -i <stor_ip_addr stor_hostname> [:port_num] [-n <iscsi_name>]] <adapter_name>	Returns parameters in discovery or send target to default inheritance behavior.
	-parameter --reset <param_name> -ip <stor_ip_addr stor_hostname> [:port_num] [-name <iscsi_name>]] <adapter_name>	

vicfg-iscsi Discovery Target Options

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

- **Dynamic discovery** – Also known as Send Targets discovery. Each time the ESX/ESXi host contacts a specified iSCSI server, it sends a Send Targets request to the server. In response, the iSCSI server supplies a list of available targets to the ESX/ESXi host.
- **Static Discovery** – The ESX/ESXi host does not have to perform any discovery. Instead, the ESX/ESXi host uses the IP addresses or domain names and iSCSI target names (IQN or EUI format names) to communicate with the iSCSI target.

The -D and -S options monitor and manage target discovery addresses. You can also use the vSphere Client to perform the same task.

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 to be seen. The iSCSI HBA must be in the same VLAN as both ports of the iSCSI array.

Listing Dynamic or Static Discovery Targets

The -l --list option lists dynamic or static discovery targets and related properties, as follows:

```
vicfg-iscsi -D -l <adapter_name>
vicfg-iscsi --discovery --list <adapter_name>
vicfg-iscsi -S -l <adapter_name>
vicfg-iscsi --static --list <adapter_name>
```

Example

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-iscsi.pl <conn_options> --discovery --list vmhba42
```

Lists dynamic discovery targets and related properties.

Adding Dynamic or Static Discovery Targets

The -a --add option adds dynamic or static discovery targets. You can perform the same operation using the vSphere Client, as discussed in the *ESX Configuration Guide* and the *ESXi Configuration Guide*. The two types of target differ as follows:

- With dynamic discovery, all storage targets associated with an IP address are discovered.
- With static discovery, you must specify the IP address and the iSCSI name of the storage target to be seen.

Use the following syntax to add a dynamic discovery target. If <portnum> is not specified, it defaults to 3260.

```
vicfg-iscsi -D -a -i <stor_ip_addr|stor_hostname> [:<portnum>] <adapter_name>
vicfg-iscsi --discovery --add --ip <stor_ip_addr|stor_hostname> [:<portnum>] <adapter_name>
```

Use the following syntax to add a static discovery target. In contrast to dynamic discovery, the iSCSI name is required.

```
vicfg-iscsi -S -a -i <stor_ip_addr|stor_hostname> [:<portnum>] -n <target_name>
               <adapter_name>
vicfg-iscsi --static --add --ip <stor_ip_addr|stor_hostname> [:<portnum>] --name <target_name>
               <adapter_name>
```

IQN Name and EUI Name

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`

When you enable software iSCSI on an ESX/ESXi host, the host generates an IQN name. 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 already 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.

Examples

The following examples require connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-iscsi.pl <conn_options> --discovery --add --ip <stor_ip_addr|stor_hostname> vmhba42
```

Adds a dynamic discovery address.

```
vicfg-iscsi.pl <conn_options> --static --add
               --ip <stor_ip_addr|stor_hostname> --name <target_name> vmhba42
```

Adds a static discovery address.

Removing Dynamic or Static Discovery Targets

The `-r --remove` option removes a dynamic or static target. You can perform the same operation using the vSphere Client, as discussed in the *iSCSI SAN Configuration Guide*.

Use the following syntax to remove a dynamic discovery target.

```
vicfg-iscsi -D -r -i <stor_ip_addr|stor_hostname> [:<portnum>] <adapter_name>

vicfg-iscsi --discovery --remove
               --ip <stor_ip_addr|stor_hostname> [:<portnum>] <adapter_name>
```

When you 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.

Use the following syntax to remove a static discovery target. You can use <stor_ip_addr> or <stor_hostname> and optionally <portnum>. If <portnum> is not specified, it defaults to 3260. You specify the required target name in IQN or EUI format. See [“IQN Name and EUI Name”](#) on page 68.

```
vicfg-iscsi -S -r -i <stor_ip_addr|stor_hostname> [:<portnum>]
-n <target_name> <adapter_name>
vicfg-iscsi --static --remove --ip <stor_ip_addr|stor_hostname> [:<portnum>]
--name <target_name> <adapter_name>
```

Example

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-iscsi.pl <conn_options> --discovery --remove --ip <stor_ip_addr|stor_hostname> vmhba42
```

vicfg-iscsi Authentication Options

This section discusses iSCSI authentication options. You can also set these options using the vSphere Client, as discussed in the *iSCSI SAN Configuration Guide*.

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

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 already established. After the discovery address is set, the new volumes to which you add a connection are exposed and can be used.

ESX/ESXi hosts support only one set of CHAP credentials per initiator. You cannot assign different CHAP credentials for different targets.

When you configure 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 ESX/ESXi host to match the credentials on the iSCSI storage.

Listing Supported and Enabled Authentication Methods

The `--authentication --list` options list supported and enabled authentication methods, as follows:

```
vicfg-iscsi -A -l <adapter_name>
vicfg-iscsi --authentication --list <adapter_name>
```

Example

The following example requires connection options. See [“Specifying Authentication Information”](#) on page 13.

```
vicfg-iscsi.pl <conn_options> --authentication --list vmhba42
```

The example first returns supported initiator authentication methods, and then supported mutual authentication methods.

Enabling Authentication

You can enable authentication on the ESX/ESXi host by specifying the authentication method and level and the user name and password to be used for authentication, as follows:

```
vicfg-iscsi -A -c <level> -m <auth_method>
-u <auth_u_name> -w <auth_password>
[-i <stor_ip_addr|stor_hostname> [:<portnum>]]
[-n <target_iscsi_name>]
<adapter_name>
```

```
vicfg-iscsi --authentication --level <level> --method <auth_method>
--auth_username <auth_u_name> --auth_password <auth_password>
[--ip <stor_ip_addr|stor_hostname> [:<portnum>]]
[-name <target_iscsi_name>]
<adapter_name>
```

For <level>, specify one of the values in [Table 4-2](#).

Table 4-2. Supported Levels for CHAP

Level	Description	Supported
chapProhibited	The host does not use CHAP authentication. If authentication is currently enabled, specify chapProhibited to disable it.	Software iSCSI Hardware iSCSI
chapDiscouraged	The host uses a non-CHAP connection, but allows a CHAP connection as fallback.	Software iSCSI
chapPreferred	The host uses CHAP if the CHAP connection succeeds, but uses non-CHAP connections as fallback.	Software iSCSI Hardware iSCSI
chapRequired	The host requires successful CHAP authentication. The connection fails if CHAP negotiation fails.	Software iSCSI

For <auth_method>, CHAP is the only valid value.

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.

Enabling Mutual (Bidirectional) Authentication

ESX/ESXi systems support mutual authentication for software iSCSI but not for hardware iSCSI.

You enable mutual authentication by using the syntax discussed in [“Enabling Authentication”](#) on page 69 but adding the -b (or --mutual) option, as follows:

```
vicfg-iscsi -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>
```

The command is not supported for hardware iSCSI.

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 only if CHAP is set to chapRequired.

For <auth_method>, CHAP is the only valid value.

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.

To enable mutual authentication

- 1 Enable authentication on the ESX/ESXi host.

```
vicfg-iscsi -A -c <level> -m <auth_method> -u <auth_u_name> -w <auth_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 ESX/ESXi host.

```
vicfg-iscsi -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>
```

- 3 Make sure that:

- The second user name and password are supported for mutual authentication on the storage side.
- CHAP and mutual CHAP use different passwords.
- CHAP is required when you set mutual CHAP to required.

Returning Authentication to Default Inheritance

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

If you use the vSphere 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.

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_auth` option, which requires the `--name` option for static discovery addresses but not for dynamic discovery targets.

```
vicfg-iscsi <conn_options> -A --reset_auth --method <auth_method>
--ip <stor_ip_addr|stor_hostname> <adapter_name>
```

```
vicfg-iscsi <conn_options> --authentication --reset_auth --method <auth_method>
--ip <stor_ip_addr|stor_hostname> --name <iscsi_name> <adapter_name>
```

vicfg-iscsi Information Retrieval Options

Information retrieval options to `vicfg-iscsi` list external HBA properties, information about targets, and LUNs.

You can also list discovery addresses and authentication information. See [“vicfg-iscsi Discovery Target Options”](#) on page 67 and [“vicfg-iscsi Authentication Options”](#) on page 69.

External HBA Properties

The `-P` option lists external (vendor-specific) properties of an iSCSI adapter, as follows:

```
vicfg-iscsi -P -l <adapter_name>
vicfg-iscsi --phba --list <adapter_name>
```

The system returns information about the vendor, model, description, and serial number of the HBA.

Target Information

The `-T` option lists target information, as follows:

```
vicfg-iscsi -T -l <adapter_name>
vicfg-iscsi --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 [“IQN Name and EUI Name”](#) on page 68.

LUN Information

The `-L` option lists LUN information, as follows:

```
vicfg-iscsi -L -l <adapter_name>
vicfg-iscsi --lun --list <adapter_name>
```

The system returns the OS device name, bus number, target ID, LUN ID, and LUN size for the LUN.

You use `-L` in conjunction with `-t` to list only LUNs on a specified target, as follows:

```
vicfg-iscsi -L -l -t <target_ID> <adapter_name>
vicfg-iscsi --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.

vicfg-iscsi Physical Network Portal Options

The `-p --pnp` option lists and sets physical network portal information.

IMPORTANT This option is supported only for hardware iSCSI.

Listing Physical Network Portal Information

The `-p -l` options list physical network portal information, as follows:

```
vicfg-iscsi -p -l <adapter_name>
vicfg-iscsi --pnp --list <adapter_name>
```

The system returns whether the MAC address is settable, the maximum and current transfer rate, the MTU size, and the MAC address.

Example

```
vicfg-iscsi.pl <conn_options> -p -l vmhba42
```

```
MAC ADDRESS SETTABLE
```

```
MAX TRANSFER RATE (Mbps)          1024
```

```
CURRENT TRANSFER RATE (Mbps)      1024
```

```
MAX FRAME SIZE                    1500
```

```
MAC ADDRESS                       00:c0:dd:0a:94:13
```

Setting Physical Network Portal Information

The `-p -M` options set the maximum transmission unit (MTU) value of the iSCSI HBA.

```
vicfg-iscsi -p -M <mtu_size> <adapter_name>
vicfg-iscsi --pnp --mtu <mtu-size> <adapter_name>
```

Example

```
vicfg-iscsi.pl <conn_options> -p -M 1500 vmhba42
```

vicfg-iscsi Network Options

The `-N` option monitors and manages iSCSI properties.

IMPORTANT This option is supported only for hardware iSCSI.

Listing Network Properties

The `-N -l` options list network information, as follows:

```
vicfg-iscsi -N -l <adapter_name>
vicfg-iscsi --network --list <adapter_name>
```

The system returns information about the IP address, subnet mask, default gateway, primary DNS, and alternative DNS.

Setting Network Properties

The `-i`, `-s`, and `-g` options set network properties.

- `-i --ip` – Sets the IPv4 address for the HBA
- `-s --subnetmask` – Sets the subnet mask for the HBA
- `-g --gateway` – Sets the default gateway for the HBA

The `-i` option sets the HBA IPv4 address, as follows:

```
vicfg-iscsi -N -i <stor_ip_addr> <adapter_name>
vicfg-iscsi --network --ip <stor_ip_addr> <adapter_name>
```

The `-s` option sets the HBA network mask, as follows:

```
vicfg-iscsi -N -s <subnet_mask> <adapter_name>
vicfg-iscsi --network --subnetmask <subnet_mask> <adapter_name>
```

The `-g` option sets the HBA default gateway, as follows:

```
vicfg-iscsi -N -g <default_gateway> <adapter_name>
vicfg-iscsi --network --gateway <default_gateway> <adapter_name>
```

You can set the IP address, subnet mask, and default gateway in one command, as follows:

```
vicfg-iscsi -N -i <ip_addr> -s <subnet mask> -g <default_gateway> <adapter_name>
vicfg-iscsi --network --ip <ip_addr> --subnetmask <subnet_mask>
--gateway <default_gateway> <adapter_name>
```

vicfg-iscsi Initiator Options

ESX/ESXi systems use a software-based iSCSI initiator in the VMkernel to connect to storage. Initiator options list information about the iSCSI initiator and set the iSCSI name and alias for the initiator.

Listing Initiator Properties

The `-I -l` options list information about the iSCSI initiator, as follows:

```
vicfg-iscsi -I -l <adapter_name>
vicfg-iscsi --iscsiname --list <adapter_name>
```

Example

```
vicfg-iscsi.pl <conn_options> -I -l vmhba42
```

Returns the iSCSI name, alias name, and alias settable bit.

Setting Initiator iSCSI Name or Alias Name

The `-I -n` and `-I -a` options set the iSCSI name or alias for the iSCSI initiator as follows:

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

Example

```
vicfg-iscsi.pl <conn_options> -I -a acme-pdp242.dept.comp.com vmhba42
```

Sets the iSCSI alias for vmhba42 to acme-pdp242.dept.comp.com.

vicfg-iscsi iSCSI Parameter Options

The `-W` option lists and sets iSCSI parameter options.

Listing iSCSI Parameter Options

The `-W -l` options list iSCSI parameter options for the HBA.

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.

```
vicfg-iscsi -W -l [-i <stor_ip_addr|stor_hostname> [:<portnum>] [-n <iscsi_name>]]
               <adapter_name>
vicfg-iscsi --parameter --list
               [--ip <stor_ip_addr|stor_hostname> [:<portnum>] [--name <iscsi_name>]] <adapter_name>
```

Listing iSCSI Parameter Option Details

The `-W -l -k` options list iSCSI parameters and whether they are settable.

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.

```
vicfg-iscsi -W -l -k
               [-i <stor_ip_addr|stor_hostname> [:<port_num>] [-n <iscsi_name>]] <adapter_name>
vicfg-iscsi --parameter --list --detail
               [--ip <stor_ip_addr|stor_hostname> [:<port_num>] [--name <iscsi_name>]] <adapter_name>
```

Setting iSCSI Parameter Options

The `-W -j` options set iSCSI parameter options for the HBA.

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.

```
vicfg-iscsi -W -j <name>=<value>
               -i <stor_ip_addr|stor_hostname> [:<port_num>] [-n <iscsi_name>]] <adapter_name>
vicfg-iscsi --parameter --set <name>=<value>
               --ip <stor_ip_addr|stor_hostname> [:<port_num>] [- --name <iscsi_name>]] <adapter_name>
```

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>”).

Table 4-3 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.

Table 4-3. Settable iSCSI Parameters

Parameter	Description
<code>dataDigestType</code>	Enabling header digest increases data integrity. When enabled, the system performs a checksum over iSCSI PDU data. Valid values are <code>digestProhibited</code> , <code>digestDiscouraged</code> , <code>digestPreferred</code> , or <code>digestRequired</code> . Corresponds to the <code>data_digest</code> and <code>header_digest</code> parameters in service console commands.
<code>MaxOutstandingR2T</code>	Max Outstanding R2T defines the R2T (Ready to Transfer) PDUs that can be in transition before an acknowledgement PDU is received. Corresponds to the <code>max_outstadbubg_r2t</code> parameter in service console commands.
<code>FirstBurstLength</code>	Maximum amount of unsolicited data an iSCSI initiator can send to the target during the execution of a single SCSI command, in bytes. Corresponds to the <code>first_burst_len</code> parameter in service console commands.
<code>MaxBurstLength</code>	Maximum SCSI data payload in a Data-In or a solicited Data-Out iSCSI sequence, in bytes. Corresponds to the <code>max_burst_len</code> parameter in service console commands.
<code>MaxRecvDataSegLen</code>	Maximum data segment length, in bytes, that can be received in an iSCSI PDU. Corresponds to the <code>max_rev_data_seg_len</code> parameter in service console commands.

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 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>
```

vicfg-iscsi Software iSCSI Options

The `-E` option monitors and manages software iSCSI.

Listing Software iSCSI Loading Status

Some setup is required before you can successfully enable iSCSI software on your system. The `-E -l` option lists software iSCSI loading status, as follows:

```
vicfg-iscsi -E -l
vicfg-iscsi --swiscsi --list
```

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

Enabling Software iSCSI

The `-E -e` option enables software iSCSI as follows:

```
vicfg-iscsi -E -e
vicfg-iscsi --swiscsi --enable
```

The system enables software iSCSI for the specified host and prints `Enabling software iSCSI`.

Disabling Software iSCSI

The `-E -d` option disables software iSCSI, as follows:

```
vicfg-iscsi -E -d
vicfg-iscsi --swiscsi --disable
```

The system disables software iSCSI and prints `Disabling software iSCSI for next boot`.

Tasks Required for Software iSCSI Setup

This section lists the tasks required for software iSCSI setup. It does not give step by step instruction. For each task, see the discussion of the corresponding command-line option in this chapter for more information.

- 1 Determine the HBA type and retrieve the HBA ID by running `vicfg-iscsi -H -l`.

```
vicf-iscsi.pl --adapter --list
```
- 2 Enable software iSCSI for the HBA by running `vicfg-iscsi -E -e`.

```
vicfg-iscsi.pl --swiscsi --enable
```
- 3 (Optional) Set the iSCSI name and alias by running `vicfg-iscsi -I`.

```
vicfg-iscsi -I -n <iscsi_name> <adapter_name>
vicfg-iscsi --iscsiname --name <iscsi_name> <adapter_name>
vicfg-iscsi -I -a <alias_name> <adapter_name>
vicfg-iscsi --iscsiname --alias <alias_name> <adapter_name>
```
- 4 Add a dynamic discovery address by running `vicfg-iscsi -D` or a static discovery address by running `vicfg-iscsi -S`.

```
vicfg-iscsi -D -a -i <stor_ip_addr|stor_hostname> [:<portnum>] <adapter_name>
vicfg-iscsi --discovery --add --ip <stor_ip_addr|stor_hostname> [:<portnum>] <adapter_name>

vicfg-iscsi -S -a -i <stor_ip_addr|stor_hostname> [:<portnum>] --n <target_name>
<adapter_name>
vicfg-iscsi --static --add --ip <stor_ip_addr|stor_hostname> [:<portnum>] --name
<target_name> <adapter_name>
```
- 5 (Optional) Set the authentication information for CHAP by running `vicfg-iscsi -A`.

```
vicfg-iscsi -A -c <level> -m <auth_method> -u <auth_u_name> -w <auth_password>
[-i <stor_ip_addr|stor_hostname> [:<portnum>] [-n <iscsi_name>]] <adapter_name>
vicfg-iscsi --authentication --level <level> --method <auth_method>
--auth_username <auth_u_name> --auth_password <auth_password>
[--ip <stor_ip_addr|stor_hostname> [:<portnum>] [-name <iscsi_name>]]
<adapter_name>
```
- 6 (Optional) Set the authentication information for mutual CHAP by running `vicfg-iscsi -A` again with a different authentication user name and password and the `-b` option.
- 7 (Optional) Set iSCSI parameters by running `vicfg-iscsi -W`.
- 8 After setup is complete, call `vicfg-rescan` to rescan all storage devices.

Tasks Required for Hardware iSCSI Setup

This section lists the tasks required for hardware iSCSI setup. It does not give step by step instruction. For each task, see the discussion of the corresponding command-line option for more information.

- 1 Determine the HBA type and retrieve the HBA ID by running `vicfg-iscsi -H -l`.
`vicf-iscsi.pl --adapter --list`
- 2 Configure the hardware initiator by running `vicfg-iscsi -N`.
- 3 (Optional) Set the iSCSI name and alias by running `vicfg-iscsi -I`.
`vicfg-iscsi -I -n <iscsi_name> <adapter_name>`
`vicfg-iscsi --iscsiname --name <iscsi_name> <adapter_name>`
`vicfg-iscsi -I -a <alias_name> <adapter_name>`
`vicfg-iscsi --iscsiname --alias <alias_name> <adapter_name>`
- 4 Add a dynamic discovery address by running `vicfg-iscsi -D` or set a static discovery address by running `vicfg-iscsi -S`.
- 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.
 Mutual CHAP is not supported for 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.

Virtual Machine Management with vmware-cmd

5

This chapter discusses `vmware-cmd`, a vSphere CLI command for virtual machine management. You can use `vmware-cmd` for registering a virtual machine on the local server, getting the power state of a virtual machine, setting configuration variables, and some other tasks.

IMPORTANT Older service console versions of `vmware-cmd` support a set of connection options and general options that differ from the options in other vSphere CLI commands. The `vmware-cmd` vSphere CLI command supports these options. The vSphere CLI command also supports the standard vSphere CLI `--server`, `--username`, and `--password` options, but none of the other connection options.

This chapter includes the following topics:

- [“Connection Options for vmware-cmd”](#) on page 79
- [“General Options for vmware-cmd”](#) on page 80
- [“Server Options for vmware-cmd”](#) on page 80
- [“Virtual Machine Options for vmware-cmd”](#) on page 80
- [“vmware-cmd Examples”](#) on page 82

Connection Options for vmware-cmd

The `vmware-cmd` vSphere CLI command supports only the following connection options. Other connection options are not supported.

Table 5-1. `vmware-cmd` Connection Options

Option	Description
<code>--server <host></code> <code>-H <host></code>	Target ESX/ESXi host or vCenter Server system.
<code>-T <target></code>	When you run <code>vmware-cmd</code> with the <code>-H</code> option pointing to a vCenter Server system, use <code>-T</code> to specify the ESX/ESXi host to run the command against.
<code>-O <port></code>	Alternative connection port. The default port number is 902.
<code>--username <username></code> <code>-U <username></code>	User who is authorized to log in to the host specified by <code>-H</code> or <code>-T</code> .
<code>--password <password></code> <code>-P <password></code>	Password for the user specified by <code>-U</code> .
<code>-Q <protocol></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` vSphere CLI command supports the following general options.

Table 5-2. vmware-cmd General Options

Option	Description
-h	Prints a help message, listing the options for this command.
-q	Runs in quiet mode with minimal output. The output does not display the specified operation and arguments.
-v	Runs in verbose mode.

Server Options for vmware-cmd

The following options perform operations on the ESX/ESXi host you are connected to. When specifying `vm-path`, choose 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)

Table 5-3. vmware-cmd Server Operations

Server Operation	Description
-l	Lists the registered virtual machines on the server.
-s register <vm-path> <datacenter> <resource_pool>	Registers a virtual machine specified by <vm-path> on the server. When you run against a vCenter Server system, you must specify the datacenter and the resource pool to register the virtual machine in. The default datacenter is <code>ha-datacenter</code> and the default resource pool is <code>Resources</code> . When you run against an ESX/ESXi host, you usually do not specify the resource pool and datacenter. However, if two virtual machines with the same name exist in two resource pools, you must specify the resource pool.
-s unregister <vm-path>	Unregisters a virtual machine specified by <vm-path> on the server.

Virtual Machine Options for vmware-cmd

The following options perform operations on virtual machines on the ESX/ESXi host you are connecting to.

These options are not preceded by minus (-) on the command line. There are no short options.

When specifying `vm-path`, choose 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)

Table 5-4. Commands That Apply to Any vSphere Host

Command	Description
<vm-path> answer	Prompts the user to answer a question for a virtual machine waiting for user input.
<vm-path> connectdevice <device_name>	Connects the specified virtual device to a virtual machine.
<vm-path> createsnapshot <name> <description> <quiesce> <memory>	<p>Creates a snapshot of the specified virtual machine, providing a name and description for the snapshot.</p> <ul style="list-style-type: none"> ■ If the <quiesce> flag is 1 and the virtual machine is powered on when the snapshot is taken, VMware Tools is used to quiesce the file system in the virtual machine. Quiescing a file system is a process of bringing the on-disk data of a physical or virtual computer into a state suitable for backups. This process might include such operations as flushing dirty buffers from the operating system's in-memory cache to disk, or other higher-level application-specific tasks. ■ If the <memory> flag is 1, a dump of the internal state of the virtual machine is included in the snapshot. Memory snapshots take longer to create.
<vm-path> disconnectdevice <device_name>	Disconnects the specified virtual device from the virtual machine.
<vm-path> getconfigfile	Returns a string containing the name of the virtual machine configuration file. This command fails if the virtual machine is not connected.
<vm-path> getguestinfo <variable>	<p>Retrieves the value for a GuestInfo variable.</p> <p>The variable contains the attributes of the guest operating system of a virtual machine. For example, you can run <code>vmware-cmd <vm-path> getguestinfo ip</code> to retrieve the IP address of a virtual machine.</p> <p>You usually use this command when VMware Technical Support or a VMware Knowledge Base article instruct you to do so.</p>
<vm-path> getproductinfo <proinfo>	<p>Returns information about the product, where <proinfo> is <code>product</code>, <code>platform</code>, <code>build</code>, <code>majorversion</code> (product major version number), or <code>minorversion</code> (product minor version number).</p> <p>If <code>product</code> is specified, the return value is one of the following:</p> <ul style="list-style-type: none"> ■ <code>gsx</code> – VMware Server or GSX Server ■ <code>esx</code> – VMware ESX ■ <code>embeddedESX</code> – VMware ESXi ■ <code>vpx</code> – VMware vCenter Server ■ <code>unknown</code> (unknown product type) <p>If <code>platform</code> is specified, the return value is one of the following:</p> <ul style="list-style-type: none"> ■ <code>win32-x86</code> – x86-based Windows system ■ <code>linux-x86</code> – x86-based Linux system ■ <code>vmnix-x86</code> – x86 ESX/ESXi microkernel
<vm-path> getstate	Retrieves the execution state of a virtual machine. The state can be <code>on</code> , <code>off</code> , <code>suspended</code> , or <code>unknown</code> .
<vm-path> gettoolslastactive	<p>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 the virtual machine powers on. It 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.</p> <p>The command can return one of the following:</p> <ul style="list-style-type: none"> 0 – VMware Tools are 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.
<vm-path> getuptime	Returns the uptime (in seconds) of the guest operating system on the virtual machine.
<vm-path> hassnapshot	Returns 1 if the virtual machine already has a snapshot. Returns 0 otherwise.
<vm-path> removesnapshots	Removes all snapshots belonging to the virtual machine. If no snapshot exists, does nothing.

Table 5-4. Commands That Apply to Any vSphere Host (Continued)

Command	Description
<vm-path> reset <powerop_mode>	Shuts down, and then reboots a virtual machine. The <code>powerop_mode</code> can be <code>hard</code> or <code>soft</code> . Default is <code>soft</code> . See “Soft and Hard Power Operations” on page 82.
<vm-path> revertstapshot	Reverts the virtual machine to the current snapshot. If no snapshot exists, does nothing and leaves the virtual machine state unchanged.
<vm-path> setguestinfo <variable> <value>	Writes a <code>GuestInfo</code> variable into memory. This is an advanced command. You usually use this command when VMware Technical Support or a VMware Knowledge Base article instruct you to do so.
<vm-path> start	Powers on a previously powered-off virtual machine or resumes a suspended virtual machine.
<vm-path> stop <powerop_mode>	Shuts down and powers off a virtual machine. The <code>powerop_mode</code> can be <code>hard</code> or <code>soft</code> . Default is <code>soft</code> . See “Soft and Hard Power Operations” on page 82.
<vm-path> suspend <powerop_mode>	Suspends a virtual machine. The <code>powerop_mode</code> can be <code>hard</code> or <code>soft</code> . Default is <code>soft</code> . See “Soft and Hard Power Operations” on page 82.

Soft and Hard Power Operations

When you call `vmware-cmd` to start, stop, reset, or suspend a virtual machine, you must supply a value for the `powerop_mode` flag. This section describes what happens when that value is `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 command performs the following operations:

- Stop – Attempts to shut down the guest operating system, and then powers off the virtual machine.
- Reset – Attempts to shut down the guest operating system, and then reboots the virtual machine.
- Suspend – Attempts to run a script in the guest operating system before suspending the virtual machine.

hard Power Operations

When you specify `hard` as the `powerop_mode` value, the command immediately and unconditionally powers off, resets, or suspends the virtual machine.

vmware-cmd Examples

The following examples for `vmware-cmd` require connection options. See [“Connection Options for vmware-cmd”](#) on page 79.

```
vmware-cmd.pl <conn_options> C:\home\vmware\win2000.vmx getstate
```

Retrieves the execution state of a virtual machine. `C:\home\vmware\win2000.vmx` is the path to the virtual machine configuration file.

```
vmware-cmd.pl <conn_options> -q C:\home\vmware\win2000.vmx reset hard
```

Performs a hard reset, that is, the virtual machine is reset immediately and unconditionally.

- `-q` indicates the quiet option (only the results of the operation are printed).
- `C:\home\vmware\win2000.vmx` is the path to the virtual machine configuration file.

```
vmware-cmd.pl <conn_options> D:\dir.vmx connectdevice "CD/DVD Drive 2"
```

Connects the virtual IDE device `CD/DVD Drive 2` in a Windows guest operating system.

```
vmware-cmd.pl <conn_options> /vmfs/volumes/4654bb7d-37601c39-9c29-00145e5a8a9d/padellvm2/padellvm2.vmx getproductinfo product
```

Returns, for example, `getproductinfo(product) = esx`.

Storage Path and Storage Plugin Management with esxcli

6

The `esxcli` command is available as a service console command (no authentication) and as a vSphere CLI. The command is primarily used for PSA (pluggable storage architecture) management. The *ESX Configuration Guide* and the *ESXi Configuration Guide* discuss PSA functionality in detail. Those documents explain how to use the vSphere Client to manage the PSA, the associated native multipathing plug-in (NMP) and third-party plug-ins using the vSphere Client.

[Table 6-1](#) lists the acronyms this chapter uses.

Table 6-1. Pluggable Storage Architecture 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.

The chapter includes these topics:

- [“esxcli Command Syntax”](#) on page 83
- [“Managing NMP with esxcli nmp”](#) on page 84
- [“Using esxcli swiscsi nic for NIC Binding”](#) on page 90
- [“Path Claiming with esxcli corestorage claiming”](#) on page 91
- [“Managing Claim Rules with esxcli corestorage claimrule”](#) on page 92
- [“Path Masking with esxcli corestorage claimrule”](#) on page 96

esxcli Command Syntax

The `esxcli` vSphere CLI command has the following syntax:

```
esxcli <conn_options> <namespace> <app> <cmd> [cmd options]
```

Option	Description
<conn_options>	Connection parameters for the vSphere CLI must precede all other parameters, or you must perform authentication in other ways. For example, you can perform authentication using vi-fastpass on vMA, or using environment variables. See “Specifying Authentication Information” on page 13. <code>esxcli</code> does not support the credential store.
<namespace>	Namespace. One of the following: nmp – VMware native multipathing commands. swiscsi – Commands in the software iSCSI name space. corestorage – VMware core storage commands.

Option	Description
<app>	Area within the name space to which the command applies.
<cmd>	Command to be called.
<cmd options>	Command options.

In contrast to other vSphere CLI commands, `esxcli` is not a Perl script and does not run with a `.pl` extension. Only the command options support corresponding short options, there are no short options for other elements (namespace, app, or command).

IMPORTANT You can run `esxcli` with `--server` pointing to an ESX/ESXi host, but not with `--server` pointing to a vCenter Server system.

`esxcli` does not support credential store authentication or the `--credstore` option.

Help for esxcli

Command-line help for the `esxcli` vSphere CLI is available on a per-level basis. You can call help as follows:

Command	Example	Output
<code>esxcli --help</code> <code>esxcli -?</code>	<code>esxcli --help</code> <code>esxcli -?</code>	Displays help for supported connection options.
<code>esxcli <conn_options> --help</code> <code>esxcli <conn_options> -?</code>	<code>esxcli --server S1 --help</code> <code>esxcli --server S1 -?</code>	Displays help for supported name spaces.
<code>esxcli <conn_parms> <namespace> --help</code> <code>esxcli <conn_parms> <namespace> -?</code>	<code>esxcli --server S1 nmp --help</code> <code>esxcli --server S1 nmp -?</code>	Displays help for supported apps for this namespace.
<code>esxcli <conn_options> <namespace> <app> --help</code> <code>esxcli <conn_options> <namespace> <app> -?</code>	<code>esxcli --server S1 nmp device --help</code> <code>esxcli --server S1 nmp device -?</code>	Displays help for supported commands for this app.
<code>esxcli <conn_options> <namespace> <app> <command> --help</code> <code>esxcli <conn_options> <namespace> <app> <command> -?</code>	<code>esxcli --server S1 nmp device setpolicy --help</code> <code>esxcli --server S1 nmp device setpolicy -?</code>	Displays help for supported options for this command.

Managing NMP with esxcli nmp

The NMP (Native Multipathing Plugin) is an extensible multipathing module that ESX/ESXi supports by default. You can use `esxcli nmp` 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 specific 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.

IMPORTANT The `esxcli nmp boot` option is internal use only and not discussed in this document.

Device Management with esxcli nmp device

The device option performs operations on devices currently claimed by the VMware NMP plugin.

esxcli 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, call `esxcli nmp path list`.

Table 6-2. esxcli nmp device list Options

Options	Description
--device <device> -d <device>	Filters the output of the command to show information about a single device. Default is all devices.

esxcli nmp device setpolicy

The `setpolicy` command sets the Path Selection Policy (PSP) for the specified device to one of the policies loaded on the system.

Table 6-3. esxcli nmp device setpolicy Options

Options	Description
--default -E	Sets the PSP back to the default for the SATP assigned to this device.
--device <device> -d <device>	Device to set the PSP for.
--psp <PSP> -P <PSP>	PSP to assign to the specified device. Call <code>esxcli nmp psp list</code> to display all currently available PSPs. The following PSPs are currently supported: <ul style="list-style-type: none"> ■ VMW_PSP_FIXED – Fixed path policy. If you use that policy, use <code>esxcli nmp fixed</code> to set or get the preferred path. ■ VMW_PSP_MRU – Most recently used path policy. ■ VMW_PSP_RR – Round robin path policy. See the <i>ESX Configuration Guide</i> and the <i>ESXi Configuration Guide</i> for a discussion of path policies.

Examples

```
esxcli <conn_options> nmp device setpolicy --device naa.xxx --psp VMW_PSP_FIXED
```

Sets the path policy for the specified device to VMW_PSP_FIXED.

Fixed Path Selection Policy Operations with esxcli nmp fixed

The `fixed` option gets and sets the preferred path policy for NMP devices configured to use VMW_PSP_FIXED.

esxcli nmp fixed getpreferred

The `getpreferred` command retrieves the preferred path on a specified device that is using NMP and the VMW_PSP_FIXED PSP.

Table 6-4. esxcli nmp fixed getpreferred Options

Options	Description
--device <device> -d <device>	Device for which you want to get the preferred path. This device must be controlled by the Fixed PSP.

Example

```
esxcli <conn_options> nmp fixed getpreferred --device naa.xxx
```

Returns the path configured as the preferred path for the specified device.

esxcli nmp fixed setpreferred

The `setpreferred` command sets the preferred path on a specified device that is using NMP and the fixed path policy.

Table 6-5. esxcli nmp fixed setpreferred Options

Options	Description
<code>--device <device></code> <code>-d <device></code>	Device for which you want to set the preferred path. This device must be controlled by the Fixed PSP. Use <code>esxcli nmp device --list</code> to list the policies for all devices.
<code>--path <path></code> <code>-p <path></code>	Path to set as the preferred path for the specified device.

Example

```
esxcli <conn_options> nmp fixed setpreferred --device naa.xxx --path vmhba3:C0:T5:L3
```

Sets the preferred path for the specified device to `vmhba3:C0:T5:L3`.

Path Operations with esxcli nmp path

The `path` option lists paths claimed by NMP. To show the devices, call `esxcli nmp device list`.

esxcli nmp path list

The `list` command lists paths claimed by the VMware NMP and shows the SATP and PSP information associated with each path.

Table 6-6. esxcli nmp path list Options

Options	Description
<code>--device <device></code> <code>-d <device></code>	Filter the output of this command to only show paths to a single device.
<code>--path <path></code> <code>-p <path></code>	Filter the output of this command to only show information for a single path. By default, the <code>--list</code> command displays information about all paths.

PSP Operations with esxcli nmp psp

The `psp` option performs operations on path selection policy plugins included with the VMware NMP plugin.

This option can be used to manage VMware PSPs and third-party PSPs. When used with third-party PSPs, the syntax depends on the third-party PSP implementation.

esxcli nmp psp getconfig

The `getconfig` command retrieves PSP configuration parameters. The type of PSP determines whether you specify `--device`, `--path`, or both.

Table 6-7. esxcli nmp psp getconfig Options

Options	Description
<code>--device <device></code> <code>-d <device></code>	Device you want to get PSP configuration information for. Use <code>--device</code> for PSPs that are set to <code>VMW_PSP_RR</code> , <code>VMW_PSP_FIXED</code> or <code>VMW_PSP_MRU</code> .
<code>--path <path></code> <code>-p <path></code>	Path you want to get PSP configuration information for. Use <code>--path</code> to retrieve information about a PSP that is set to <code>VMW_PSP_FIXED</code> or <code>VMW_PSP_MRU</code> . No path configuration information is available for <code>VMW_PSP_RR</code> .

Examples

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

Returns the PSP configuration for the specified device.

```
esxcli <conn_options> nmp psp getconfig --path vmhba4:C1:T2:L23
```

Returns the PSP configuration for the specified path.

esxcli nmp psp list

Shows the list of Path Selection Plugins on the system and a brief description of each.

esxcli nmp psp setconfig

This command supports future third-party PSA expansion. The `setconfig` command sets PSP configuration parameters for those third-party extensions.

NOTE Use `esxcli roundrobin setconfig` for other path policy configuration.

The options depend on the currently set path policy.

Table 6-8. esxcli nmp psp setconfig Options

Options	Description
<code>--config <config_string></code> <code>-c <config_string></code>	Configuration string to set for the device specified by <code>--path</code> . VMW_PSP_FIXED – If the current path policy is VMW_PSP_FIXED, you can specify the preferred path using the <code>preferred</code> configuration string. VMW_PSP_RR – If the current path policy is VMW_PSP_RR, you can specify a policy string. The string can be <code>iops</code> , <code>bytes</code> , or <code>rr</code> (round robin). Use this option to specify how the round robin PSP determines when to switch paths. <ul style="list-style-type: none"> ■ <code>iops =<number></code> – Switch to the next path as soon as the specified number of I/O operations has been performed. ■ <code>bytes=<number></code> – Number of bytes before path switching. ■ <code>useANO=<number></code> – If set to 1, the round robin PSP includes paths in the active, unoptimized state in the round robin set. If 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.
<code>--device <device></code> <code>-d <device></code>	Device for which you want to customize the path policy.
<code>--path <path></code> <code>-p <path></code>	Path for which you want to customize the path policy.

Examples

```
esxcli <conn_options> nmp psp setconfig --path vnhba42:C1:T1:L1 --config preferred
```

Sets the preferred path to the specified path. Supported only if the device uses VMW_PSP_FIXED.

```
esxcli <conn_options> nmp psp setconfig --device naa.xxx --config "policy=iops;iops=4200"
```

Sets the specified device to change paths each time 4200 I/O operations have been completed. The policy and the setting of `iops` are separated by a semicolon. Supported only if the device is set up to use VMW_PSP_RR.

Use `esxcli nmp roundrobin setconfig` to set the value of round robin parameters.

Round Robin Operations with esxcli nmp roundrobin

The `roundrobin` option sets round robin path options on a device controlled by the `roundrobin` PSP.

esxcli nmp roundrobin getconfig

The `getconfig` command retrieves path selection settings for a device that is using the `roundrobin` PSP.

Table 6-9. esxcli nmp roundrobin getconfig Options

Options	Description
<code>-d <device></code> <code>--device <device></code>	Device to get roundrobin properties for.

esxcli nmp roundrobin setconfig

The `setconfig` command specifies under which conditions a device that is using the `roundrobin` PSP changes to a different path. You can use `--bytes` or `--iops` to specify when the path should change.

Table 6-10. esxcli nmp roundrobin setconfig Options

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 PSP (VMW_PSP_RR).
<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. Either <code>bytes</code> or <code>iops</code> . 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.

Examples

```
esxcli <conn_options> nmp roundrobin setconfig --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> nmp roundrobin setconfig --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.

SATP Operations with esxcli nmp satp

The `satp` option manages SATPs and allows you to perform a number of tasks.

- Retrieve and set configuration parameters
- Add and delete 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

By default, the default SATP for an active-active array with a vendor and model not listed in the SATP rules is `VMW_SATP_LOCAL`.

esxcli nmp satp addrule

The `addrule` command adds a rule to the list of claim rules for the specified SATP. 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 nmp satp addrule --satp="VMW_SATP_LOCAL" --vendor="VMWARE" --model="Virtual"
--description="VMware virtual disk"
```

Table 6-11. esxcli nmp satp addrule Options

Options	Description
<code>--claim-option</code> <code>-c</code>	Claim option string to set when adding the SATP claim rule.
<code>--description</code> <code>-e</code>	Description string to set when adding the SATP claim rule.
<code>--driver</code> <code>-D</code>	Driver string to set when adding the SATP claim rule. You can specify <code>--driver</code> , <code>--vendor</code> , <code>--transport</code> , or <code>--model</code> .

Table 6-11. esxcli nmp satp addrule Options (Continued)

Options	Description
--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> . You can specify <code>--driver</code> , <code>--vendor</code> , <code>--transport</code> , or <code>--model</code> .
--option -o	Option string to set when adding the SATP claim rule.
--satp -s	SATP for which the new rule is added.
--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> . You can specify <code>--driver</code> , <code>--vendor</code> , <code>--transport</code> , or <code>--model</code> .
--vendor -V	Vendor string to set when adding the SATP claim rule. You can specify <code>--driver</code> , <code>--vendor</code> , <code>--transport</code> , or <code>--model</code> .

Examples

```
esxcli <conn_options> nmp satp addrule --satp="VMW_SATP_LOCAL" --vendor="VMWARE"
--model="Virtual" --description="VMware virtual disk"
```

Adds a rule that specifies that disks with vendor string `VMWARE` and model string `Virtual` should be added to `VMW_SATP_LOCAL`.

```
esxcli <conn_options> nmp satp addrule --satp="VMW_SATP_LOCAL" --driver="somedriver"
```

Adds a rule that specifies that disks with the driver string `somedriver` should be added to `VMW_SATP_LOCAL`.

```
esxcli <conn_options> nmp satp addrule --satp VMW_SATP_DEFAULT_AA --vendor="ABC" --model="^120"
```

Adds 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 nmp satp deleterule

The `deleterule` command deletes a rule from the list of claim rules for the specified SATP. You can call this command with the same options as `addrule`, shown in [Table 6-11](#).

Example

```
C:\WINDOWS\system32>esxcli <conn_options> nmp satp deleterule
--satp="VM_VMW_SATP_LOCAL" --vendor="VMWARE" --model="Virtual"
```

Deletes the rule that assigns devices with vendor string `VMWARE` and model string `Virtual` to `VM_VMW_SATP_LOCAL`.

esxcli nmp satp getconfig

The `getconfig` command retrieves per-path or per-device SATP configuration parameters. For each SATP, specify either `--device` or `--path`, but not both.

Table 6-12. esxcli nmp satp getconfig Options

Options	Description
--device <device> -d <device>	Device to retrieve SATP configuration for. Not all SATPs support the <code>getconfig</code> option for devices.
--path -p	Path to retrieve SATP configuration for. Not all SATPs support the <code>getconfig</code> option for paths.

esxcli nmp satp list

The `list` command lists the SATPs that are currently loaded into the NMP system and displays information about those SATPs. This command supports no options.

esxcli nmp satp listrules

The `listrules` command lists the claim rules for SATPs.

Table 6-13. esxcli nmp satp listrules Options

Options	Description
<code>--satp</code>	Displays the SATP rules for the specified SATP.
<code>-s</code>	If you use this option, it must specify a valid loaded SATP plugin, as displayed by <code>esxcli nmp satp list</code> .

esxcli nmp satp setconfig

The `setconfig` command sets per-path or per-device SATP configuration parameters. This command sets the configuration for a specified device or path, regardless of the SATP currently associated with that device or path.

This command is intended for use with third-party SATPs loaded into the system.

Table 6-14. esxcli nmp satp setconfig Options

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"> ■ <code>VMW_SATP_ALUA_CX</code> ■ <code>VMW_SATP_ALUA</code> ■ <code>VMW_SATP_CX</code> ■ <code>VMW_SATP_INV</code> You can specify one of the following device configuration strings: <ul style="list-style-type: none"> ■ <code>navireg_on</code> – starts automatic registration of the device with Navisphere. ■ <code>navireg_off</code> – stops the automatic registration of the device. ■ <code>ipfilter_on</code> – stops the sending of the host name for Navisphere registration. Used if host is known as <code>localhost</code>. ■ <code>ipfilter_off</code> – enables the sending of the host name during Navisphere registration.
<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.

esxcli nmp satp setdefaultpsp

The `setdefaultpsp` command sets the default PSP for a specified SATP.

Table 6-15. esxcli nmp satp setdefaultpsp Options

Options	Description
<code>--psp</code> <code>-P</code>	Default path selection policy to set for the SATP specified by <code>--satp</code> .
<code>--satp</code> <code>-s</code>	SATP name for the plugin for which you want to set the default PSP.

Using esxcli swiscsi nic for NIC Binding

The `esxcli swiscsi nic` command specifies NIC bindings for VMkernel NICs.

esxcli swiscsi nic add

The `add` command adds a software iSCSI NIC to the specified adapter.

Table 6-16. esxcli swiscsi nic add Options

Options	Description
--adapter -d	Device (adapter) to perform the operation on.
-n --nic	Name of the iSCSI NIC to be added or removed.

esxcli swiscsi nic remove

The `remove` command removes a VMkernel iSCSI NIC from the current iSCSI configuration.

Table 6-17. esxcli swiscsi nic remove Options

Options	Description
--adapter -d	Device (adapter) to perform the operation on.
-n --nic	Name of the iSCSI NIC to be added or removed.

esxcli swiscsi nic list

The `list` command lists existing software iSCSI NICs for the specified adapter.

Table 6-18. esxcli swiscsi nic remove Options

Options	Description
--adapter -d	Device (adapter) to perform the operation on.

Examples

```
esxcli <conn_options> swiscsi nic list -d vmhba34
```

Lists existing software iSCSI NICs on adapter vmhba34.

```
esxcli <conn_options> swiscsi nic add -n vmk2 -d vmhba34
```

Adds a software iSCSI NIC called vmk2 to adapter vmhba34.

```
esxcli --server 10.NN.NNN.NNN swiscsi nic remove -n vmk2 -d vmhba34
```

Removes the software iSCSI NIC vmk2 from adapter vmhba34.

Path Claiming with esxcli corestorage claiming

The `esxcli corestorage claiming` option 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 events fail and reclaiming also does not work.

IMPORTANT The `--help` command for `esxcli corestorage claiming` includes the `autoclaim` command. Do not use this command unless instructed to do so by VMware support staff.

esxcli corestorage claiming reclaim

The `reclaim` troubleshooting command first attempts to unclaim all paths to a device. The command then runs the loaded claim rules on each of the unclaimed paths to reclaim those paths. It is normal for this command to not succeed if the device is in use. Only PSA plugin developers or administrators who troubleshoot PSA plugins use this command.

Table 6-19. esxcli corestorage claiming reclaim Options

Options	Description
--device <device> -d <device>	Name of the device on which all paths are reclaimed.
--help	Displays the help message.

esxcli corestorage claiming unclaim

The `unclaim` command unclaims a path or set of paths, and disassociates those paths from a PSA plugin. It is normal for this commands to not succeed if the device is in use.

You can only unclaim active paths with no I/O. 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.

Table 6-20. esxcli corestorage claiming unclaim Options

Options	Description
--adapter <adapter> -A <adapter>	If --type is set to <code>location</code> , specifies the name of the HBA for the paths you want to unclaim. If you do not specify this option, unclaiming runs on paths from all adapters.
--channel <channel> -C <channel>	If --type is set to <code>location</code> , specifies the SCSI channel number for the paths you want to unclaim. If you do not specify this option, unclaiming runs on paths from all channels.
--device <device> -d <device>	If --type 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.
--driver <driver> -D <driver>	If --type is <code>driver</code> , unclaims all paths specified by this HBA driver.
--lun <lun_number> -L <lun_number>	If --type is <code>location</code> , specifies the SCSI LUN for the paths to unclaim. If you do not specify --lun, unclaiming runs on paths with any LUN number.
--path <path> -p <path>	If --type is <code>path</code> , unclaims a path specified by its path UID or runtime name.
--plugin <plugin> -P	If --type is <code>plugin</code> , unclaims all paths for a specified multipath plugin. <plugin> can be any valid PSA plugin on the system. By default only NMP and MASK_PATH are available, but additional plugins might be installed.
--target <target> -T <target>	If --type is <code>location</code> , unclaims the paths with the SCSI target number specified by target. If you do not specify --target, unclaiming runs on paths from all targets.
--type <type> -t <type>	Type of unclaim operation to perform. Valid values are <code>location</code> , <code>path</code> , <code>driver</code> , <code>device</code> , and <code>plugin</code> .

Example

```
esxcli <conn_options> corestorage claiming unclaim --type location -A vmhba1
```

Troubleshooting command to unclaim all paths to vmhba1. Run `vicfg-mpath.pl <conn_options> -l` to verify the command succeeded.

Managing Claim Rules with esxcli corestorage claimrule

The PSA uses claim rules to determine which multipathing module should claim the paths to a particular device and to manage the device. `esxcli corestorage claimrule` manages claim rules.

Claim rule modification does not operate on the VMkernel directly. Instead it operates on the configuration file by adding and removing rules.

Changing the current claim rules in the VMkernel requires these calls:

- 1 A call to one of the `esxcli corestorage claimrule` modification commands (`add`, `remove`, or `move`).
- 2 A call to `esxcli corestorage claimrule load` to replace the current rules in the VMkernel with the modified rules from the configuration file.

esxcli corestorage claimrule add

The `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. See [“Path Masking with esxcli corestorage claimrule”](#) on page 96.

Table 6-21. `esxcli corestorage claimrule add` Options

Options	Description
<code>--adapter <adapter></code> <code>-A <adapter></code>	Adapter of the paths to use. Valid only if <code>--type</code> is <code>location</code> .
<code>--channel <channel></code> <code>-C <channel></code>	Channel of the paths to use. Valid only if <code>--type</code> is <code>location</code> .
<code>--driver <driver></code> <code>-D <driver></code>	Driver for the HBA of the paths to use. Valid only if <code>--type</code> is <code>vendor</code> .
<code>--force</code> <code>-f</code>	Force claim rules to ignore validity checks and install the rule.
<code>--lun <lun_number></code> <code>-L <lun_number></code>	LUN of the paths to use. Valid only if <code>--type</code> is <code>location</code> .
<code>--model <model></code> <code>-M <model></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.pl <conn_options> -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. 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 ESX/ESXi host. See “Path Masking with esxcli corestorage claimrule” on page 96.
<code>--rule <rule_ID></code> <code>-r <rule_ID></code>	Rule ID to use. Run <code>esxcli corestorage 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 <target></code> <code>-T <target></code>	Target of the paths to use. Valid only if <code>--type</code> is <code>location</code> .
<code>--transport <transport></code> <code>-R <transport></code>	Transport of the paths to use. Valid only if <code>--type</code> is <code>transport</code> . Valid values are <ul style="list-style-type: none"> ■ <code>block</code> – block storage connection ■ <code>fc</code> – FibreChannel transmission ■ <code>iscsivendor</code> – iSCSI connection ■ <code>iscsi</code> – not currently used ■ <code>ide</code> – IDE storage connection ■ <code>sas</code> – SAS storage connection ■ <code>sata</code> – SATA storage connection ■ <code>usb</code> – USB storage connection ■ <code>parallel</code> – parallel transmission ■ <code>unknown</code>

Table 6-21. esxcli corestorage claimrule add Options (Continued)

Options	Description
--type <type> -t <type>	Type of matching to use for the operation. Valid values are vendor, location, driver, and transport.
--vendor -V	Vendor of the paths to use. Valid only if --type is vendor. Valid values are values of the Vendor string from the SCSI inquiry string. Run <code>vicfg-scsidevs.pl <conn_options> -l</code> on each device to see vendor string values.

Examples

esxcli <conn_options> corestorage claimrule add -r 321 -t location -A vmhba0 -C 0 -T 0 -L 0 -P NMP

Adds rule 321, which claims the path on adapter vmhba0, channel 0, target 0, LUN 0 for the NMP plugin.

esxcli <conn_options> corestorage claimrule add -r 429 -t driver -D mptscsi -P MASK_PATH

Adds rule 429 to claim all paths provided by an adapter with the mptscsi driver for the MASK_PATH plugin.

esxcli <conn_options> corestorage claimrule add -r 914 -t vendor -V VMWARE -M Virtual -P NMP

Adds rule 914 to claim all paths with a vendor string that matches VMWARE and a model string Virtual for the NMP plugin.

esxcli <conn_options> corestorage claimrule add -r 1015 -t transport -R fc -P NMP

Adds rule 1015 to claim all paths provided by Fibre Channel adapters for the NMP plugin.

esxcli corestorage claimrule convert

The `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.

You use this command as follows:

- 1 Call `esxcli corestorage claimrule convert` without options. That call displays the list of claim rules that 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 Call `esxcli corestorage claimrule convert --commit` to actually commit the change.

When you convert LUN masking to the claim rule format after an ESX/ESXi upgrade from ESX/ESXi 3.5 to ESX/ESXi 4.0, 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 is not supported for software iSCSI LUNs.

Inspect the list of generated claim rules carefully before you commit them by using `--commit`.

Table 6-22. esxcli corestorage claimrule convert Options

Options	Description
--commit -C	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.

esxcli corestorage claimrule delete

The `delete` command deletes 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 delete this rule, unless you want to unmask these devices.

Table 6-23. esxcli corestorage claimrule delete Options

Options	Description
<code>--rule <rule_ID></code> <code>-r <rule_ID></code>	ID of the rule to be deleted. Run <code>esxcli corestorage claimrule list</code> to see the rule ID.

Example

```
esxcli <conn_options> corestorage claimrule delete -r 1015
```

Deletes rule 1015.

esxcli corestorage claimrule list

The `list` command lists all claim rules on the system. This command takes no options.

Example

```
esxcli <conn_options> corestorage claimrule list
```

Might display something like the following:

Rule	Class	Type	Plugin	Matches
0	runtime	transport	NMP	transport=usb
1	runtime	transport	NMP	transport=sata
2	runtime	transport	NMP	transport=ide
3	runtime	transport	NMP	transport=block
101	runtime	vendor	MASK_PATH	vendor=<VENDOR> model =<model>
101	file	vencor	MASK_PATH	vendor=<VENDOR> model =<model>
55535	runtime	vendor	NMP	vendor=* model=*

esxcli corestorage claimrule load

The `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`.

esxcli corestorage claimrule move

The `move` command moves a claim rule from one rule ID to another.

Table 6-24. esxcli corestorage claimrule move Options

Options	Description
<code>--new-rule <rule_ID></code> <code>-n <rule_ID></code>	New rule ID you want to give to the rule specified by the <code>--rule</code> option.
<code>--rule <rule_ID></code> <code>-r <rule_ID></code>	ID of the rule to be deleted. Run <code>esxcli corestorage claimrule list</code> to display the rule ID.

Example

```
esxcli <conn_options> corestorage claimrule move -r 1015 -n 1016
```

Renames rule 1016 to rule 1015. Deletes rule 1016.

esxcli corestorage claimrule run

The run command runs path claiming rules. This command is for troubleshooting and boot time configuration.

Table 6-25. esxcli corestorage claimrule run Options

Options	Description
--adapter <adapter> -A <adapter>	If --type is location, 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.
--channel <channel> -C <channel>	If --type is location, 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.
--lun <lun> -L <lun>	If --type is location, 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.
--path <path_UID> -p <path_UID>	If --type is path, this option indicates the unique path identifier (UID) or the runtime name of a path to run claim rules on.
--target <target> -T <target>	If --type is location, 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.
--type <location path all> -t <location path all>	Type of claim to perform. By default, uses all, which means claim rules run without restriction to specific paths or SCSI addresses. Valid values are location, path, and all.
--wait -w	If this 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. You can use this option only if you also use --type all.

Path Masking with esxcli corestorage claimrule

With ESX/ESXi 4.0, you use the MASK_PATH plugin instead of an advanced configuration option to mask paths.

Masking a path requires these calls:

- 1 A call to `esxcli corestorage claimrule list`.

You run this command to see what the next available rule ID is. User rule IDs start at 101. If this command shows that rule 101 and 102 already exist, you can specify 103 for the rule to add.

The claim rules are evaluated in numerical order starting from 0.

- Rules 0 - 100 are reserved for internal use by VMware.
- Rules 101 - 65435 are available for general use.
- Rules 65436 - 65535 are reserved for internal use by VMware.
- Other 3rd party supplied Multipathing Plugins installed on your system might use claim rules in the range of 101 - 65435.

When adding or deleting claim rules, be sure to work with rules in the correct numeric range. When you add a MASK_PATH claimrule, choose a rule with a rule ID lower than the rule ID that causes NMP or some other multipathing plugin to claim the path.

- 2 A call to `esxcli corestorage claimrule add --plugin MASK_PATH --rule <ruleID> --type <type> -A <adapter>`.

- 3 A call to `esxcli corestorage claimrule list` to verify that the claimrule was added correctly.
- 4 A call to `esxcli corestorage claimrule load` to load the path claiming rules.
- 5 A call to `corestorage claiming unclaim` for each path to the masked device to remove the old rules.
- 6 A call to `esxcli corestorage claimrule run` to run the path claiming rules, which include the newly added rules.

Unmasking a path requires these calls:

- 1 A call to `esxcli corestorage claimrule delete` to remove the MASK_PATH claimrule.
- 2 A call to `esxcli corestorage claimrule list` to verify that the claim rule was deleted correctly.
- 3 A call to `esxcli corestorage claimrule load` to reload the path claiming rules from the configuration file into the VMkernel.
- 4 A call to `esxcli corestorage claiming unclaim` for each path to the masked device. For example:
`esxcli corestorage claiming unclaim -t location -A vmhba0 -C 0 -T 0 -L 149`
- 5 A call to `esxcli corestorage claimrule run` to run the path claiming rules.

Examples

```
esxcli <conn_options> corestorage claimrule add --plugin MASK_PATH --rule 103
--type location -A vmhba0 -C 0 -T 0 -L 3
```

Adds rule 103 to mask path `vmhba0:C0:T0:L3`.

```
esxcli <conn_options> corestorage claimrule add --plugin MASK_PATH --rule 103
--type location -A vmhba0 -C 0 -L 3
```

Adds rule 103 to mask all paths to LUN 3 on `vmhba0` and channel 0.

```
esxcli <conn_options> corestorage claimrule add --plugin MASK_PATH --rule 103
--type vendor -V ACME -M SuperDisk
```

Adds rule 103 to mask all paths to a LUN with vendor ACME and model SuperDisk.

Appendix: Commands for Managing Legacy Servers

Most vSphere CLI commands are supported against both ESX 3.5 Update 2 and later and against ESX 4.x systems. The two exceptions, `vicfg-mpath` and `vihostupdate` are supported only against ESX 4.x system. For those two commands, the 3.5 versions are included in the vSphere CLI package as `vicfg-mpath35` and `vihostupdate35`.

This appendix is a reference to `vicfg-mpath35` and `vihostupdate35`. The information is equivalent to the `vicfg-mpath` and `vihostupdate` documentation in the documentation set for VMware Infrastructure 3.5.

The appendix includes these topics:

- [“Storage Array Configuration for VMware Infrastructure 3.5 Systems with vicfg-mpath35”](#) on page 99
- [“Performing Maintenance on VMware Infrastructure 3.5 Systems with vihostupdate35”](#) on page 101

Storage Array Configuration for VMware Infrastructure 3.5 Systems with vicfg-mpath35

The `vicfg-mpath35` command can configure multipath settings for Fibre Channel or iSCSI LUNs. For more information on multipathing, see the *ESX Server Configuration Guide* and the *ESX Server 3i Configuration Guide* for version 3.5. The *Fibre Channel SAN Configuration Guide* and the *iSCSI SAN Configuration Guide* discuss some storage-specific aspects of multipathing.

Names of virtual machine HBAs are not guaranteed to be valid across reboots. Use VML LUN names to be sure of consistency. The VML name of a LUN is the unique name given to that LUN by VMware. This name is unique to a LUN and remains associated with a LUN across reboots.

Options

[Table A-1](#) lists the command-specific options for `vicfg-mpath35`.

IMPORTANT If you are changing the preferred path or if you change the state of a path:

- 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.
-

You must also specify connection options. Run `vicfg-mpath35` for a list of supported connection options.

Table A-1. Options for `vicfg-mpath35`

Option	Description
<code>--bulk</code> <code>-b</code>	Shows all LUNs and paths in a format that scripts can parse easily.
<code>--detailed</code> <code>-d</code>	Shows all information about a LUN and its paths including the VML name of the LUN. A LUN's VML name is a unique name VMware assigns to the LUN. This name is globally unique to a LUN and remains associated with a LUN across reboots.

Table A-1. Options for vicfg-mpath35 (Continued)

Option	Description
--hbas -a	Prints the list of HBAs that are identifiable by a unique ID. This includes Fibre Channel and iSCSI devices. Parallel and block devices do not appear in this list.
--list -l	Lists all LUNs on the system and the paths to these LUNs through adapters. For each LUN, this command displays the type, internal name, console name, size, paths, and path selection policy.
--lun=<lun> -L=<lun>	Required to specify the LUN to use in operations. This option is a required parameter for other options and is not used by itself.
--path=<path> -P=<path>	Required to specify the path to use in operations. This option is a required parameter for other options and is not used by itself. The path name can be either the long path UID or the shorter runtime name of the path. You can use --path to filter what the list commands return.
--policy [mru fixed] -p [mru fixed]	Set the policy for a given LUN to mru or fixed. This option requires the --lun option. <ul style="list-style-type: none"> ■ Most Recently Used (mru) selects the most recently used path to send I/O to a device. ■ Fixed (fixed) uses only the active path. Two additional policies, round robin (rr) and custom are available on an experimental basis. See the technical note <i>Round Robin Load Balancing</i> .
--preferred -f	Set the specified path to be the preferred path for a specified LUN. When you set this option, you must also set the --lun and --path options.
--query -q	Query a specific LUN for its information and print the information. When you set this option, you must also set the --lun option.
--state [on off] -s [on off]	Set the state of a given LUN path to either on or off. This option requires that both the --lun and --path options are also set.
--vihost <host> -h <host>	When you run a Remote CLI with the --server option pointing to a vCenter Server system, use --vihost to specify the ESX/ESXi system host to run the command against.

Examples

The following examples for vicfg-mpath35 require connection options. Run vicfg-mpath35 --help for a list of supported connection options.

```
vicfg-mpath35.pl <conn_options> -l
```

Displays all available paths.

```
vicfg-mpath35.pl <conn_options> -q --lun=vm1.123456
```

Displays the paths for disk vm1.123456.

```
vicfg-mpath35.pl <conn_options> --policy=mru --lun=vmhba0:0:1
```

Sets the path policy for disk vmhba0:0:1 to mru.

```
vicfg-mpath35.pl <conn_options> --path=vmhba1:0:1 --lun=vmhba0:0:1 --state=on
```

Enables a path for disk vmhba0:0:1.

```
vicfg-mpath35.pl <conn_options> --path=vmhba0:1:1 --state=off --lun=vmhba0:0:1 -p fixed
```

Disables a path and sets the policy to fixed for disk vmhba0:0:1.

```
vicfg-mpath35.pl <conn_options> -l
```

Lists all LUNs on the system and the paths to these LUNs through adapters.

```
vicfg-mpath35.pl <conn_options> -a
```

Displays LUN information. Results in output like the following.

```
vmhba2 2305843973628581845 42:2.0
vmhba3 2305843973628747050 4c:00.0
vmhba4 2306125448607554858 4c:00.1
vmhba5 50:1.1
```

```
vicfg-mpath35.pl <conn_options> -b
```

Returns a bulk path list suitable for parsing.

```
vicfg-mpath35.pl <conn_options> --policy fixed --path vmhba2:0:1 --lun vmhba2:0:1 --preferred
```

Sets the preferred path. Results in output like the following:

```
Setting vmhba2:0:1 -- vmhba2:0:1 as preferred path
Setting vmhba2:0:1 policy to fixed
```

Performing Maintenance on VMware Infrastructure 3.5 Systems with vihostupdate35

The `vihostupdate35` command performs maintenance of your ESX Server 3i hosts. The command can install software updates, enforce software update policies, and track installed software.

IMPORTANT The `vihostupdate35` command is supported only on ESX Server 3i. For maintenance of your ESX Server 3 version 3.5 host, use the `esxupdate` command. This command is not supported on vCenter Server systems.

Software updates might be patches for addressing critical security issues or urgent bug fixes, or they might be general updates or maintenance releases. Updates might also be updates to third-party drivers, modules, or CIM providers. They might be located on the local file system or on an NFS, FTP, or HTTP server. Each update consists of a descriptor file and a set of packages. The descriptor controls the installation process and checks that requirements are met. For example, you might be required to power off all virtual machines running on the server you are about to update, or you might have to reboot the server after the update.

For a detailed discussion of `vihostupdate35`, see the *ESX Server 3i Setup Guide*.

To update the firmware of an ESX Server 3i host

- 1 Download the update bundle to the machine where you run the `vihostupdate35` command.
- 2 Call the `vihostupdate35` command.

The `vihostupdate35` options either work with the bundle directly, or point to a metadata file.

The update process first pushes the update bundle to the host, and then requests that the host perform the update.

Options

[Table A-2](#) lists the command-specific options for `vihostupdate35`. You must also specify connection options. Run `vihostupdate35` for a list of supported connection options.

Table A-2. Options for `vihostupdate35`

Option	Description
<code>--bundle <bundle_file_name></code> <code>-b <bundle_file_name></code>	Unpack the downloaded bundle ZIP file. If you specify this option, you cannot specify <code>--metadata</code> .
<code>--install</code> <code>-i</code>	Patch the host with applicable packages in the update bundle. This option takes no arguments, but you must also include either <code>-b</code> to specify a bundle, or <code>-m</code> to specify a metadata file.
<code>--metadata <metadata_xml_file></code> <code>-m <metadata_xml_file></code>	Path to the <code>metadata.xml</code> file that contains information about the update bundle. If you specify this option, you cannot specify <code>--bundle</code> .
<code>--query</code> <code>-q</code>	List packages installed on the host. This option returns version information for the ESXi host and for all installed packages.
<code>--scan <dir></code> <code>-s <dir></code>	Scan for packages that apply to the host in the directory specified by <code><dir></code> .

Examples

The following examples for `vihostupdate35` require connection options. Run `vihostupdate35` for a list of supported connection options.

Assume the path to an update bundle is `EESX-142-test-release.zip`. If you then go to the directory where you copied that bundle, you can run the following commands:

```
vihostupdate35.pl <conn_options> -i -b EESX-142-test-release.zip
```

Unpacks and patches the host

```
vihostupdate35.pl <conn_options> -b EESX-142-test-release.zip
```

Unpacks the file, but does not patch the host.

```
vihostupdate35.pl <conn_options> -i -m EESX-142-test-release.zip/metadata.xml
```

Patches a host using the file `EESX-142-test-release.zip`.

Running the first example is equivalent to running the second and third in sequence.

Index

Numerics

2gbsparse disk format 57

A

authentication

 default inheritance 71

 information 13

 iSCSI 69

 methods 69

 mutual, iSCSI 70

 returning to default inheritance 71

B

backing up configuration information 39

bulletins 42

bundles, third-party 44

C

claim rules 92

 adding 93

 converting 94

 deleting 95

 from 3.5 systems 94

 listing 95

 loading 95

 moving 95

 running 96

cloning virtual disks 59

command-line connection parameters 15

commands with esxcfg prefix 19

configuration backup 39

configuration file for authentication 15

configuration file path 45

configuration files 15

configuration information 39

connection options 13

copying files 51

cp936 encoding 17

creating directories 51

creating pass-through RDM 61

creating session files 14

creating VMFS 56

credential store

 esxcli 84

 precedence 14

D

default inheritance 71, 75

deleting virtual disks 59

depots 42, 44

device management 85

device target 55

DHCP 31

diagnostic partitions

 activating 26

 deactivating 26

 managing 25

 vicfg-dumppart 25

disabling software iSCSI 76

discovery targets 67

 adding 67

 listing 67

disk attributes 56

disk formats supported by vmkfstools 57

disk partition target 55

disks See Also virtual disks

disks, cloning 59

displaying virtual disks 62

distributed virtual switches 36

DNS configuration 31

downloading files 51

downloading vSphere CLI package 11

dump partitions, vicfg-dumppart 25

DVS port groups 36

dynamic discovery targets 67

 adding 67

 listing 67

 removing 68

E

eagerzeroedthick disk format 57

enabled authentication methods 69

enabling iSCSI authentication 69

enabling mutual iSCSI authentication 70

enabling software iSCSI 76

encoding

 cp936 17

 ISO-8859-1 17

 Shift_JIS 17

environment variables for authentication 15

esxcfg prefix 19

esxcfg-mpath35 99

- esxcfg-nas **28**
- esxcfg-nics **32**
- esxcfg-rescan **30**
- esxcfg-snmp **40**
- esxcfg-vmhbadevs **29**
- esxcli
 - command syntax **83**
 - credential store authentication **84**
 - help **84**
- esxcli corestorage claiming reclaim **91**
- esxcli corestorage claiming unclaim **92**
- esxcli corestorage claimrule
 - add command **93**
 - convert command **94**
 - delete command **95**
 - list command **95**
 - load command **95**
 - move command **95**
 - run command **96**
- esxcli nmp device
 - list command **85**
 - setpolicy command **85**
- esxcli nmp fixed
 - getpreferred command **85**
 - setpreferred command **86**
- esxcli nmp path list command **86**
- esxcli nmp psp **86**
 - getconfig command **86**
 - list command **87**
 - setconfig command **87**
- esxcli nmp roundrobin **87**
 - getconfig command **87**
 - setconfig command **88**
- esxcli nmp satp **88**
 - addrule command **88**
 - deleterule command **89**
 - getconfig command **89**
 - list command **89**
 - lstrules command **90**
 - setconfig command **90**
 - setdefaultpsp command **90**
- esxcli swiscsi nic **90**
 - add command **90**
 - list command **91**
 - remove command **91**
- eui.xxx device name **27**
- examples
 - cloning raw disk **60**
 - cloning virtual disk **60**
 - deleting a virtual disk **59**
 - extending virtual disks **60**
 - extending VMFS volume **57**

- inflating virtual disks **59**
- initializing virtual disks **58**
- iSCSI storage setup **76**
- listing attributes **56**
- physical compatibility mode RDM **61**
- renaming virtual disks **59**
- svmotion **46**
- vicfg-cfgbackup **40**
- vicfg-dns **31**
- vicfg-dumppart **26**
- vicfg-module **47**
- vicfg-mpath **27**
- vicfg-mpath35 **100**
- vicfg-nas **28**
- vicfg-nics **32**
- vicfg-ntp **33**
- vicfg-rescan **30**
- vicfg-route **34**
- vicfg-snmp **41**
- vicfg-syslog **42**
- vicfg-user **49**
- vicfg-vmknics **36**
- vicfg-vswitch **38**
- vifs **52**
- virtual compatibility mode RDM **61**
- virtual disks **58**
- VMFS file system **56**
- execution options **17**
- extendfs **53**
- extending file system partition, vmkfstools **57**
- extending virtual disks **60**
- extending volumes **57**
- external HBA properties **71**

F

- Fibre Channel LUNs **27**
- Fibre Channel storage **99**
- file management, vifs **50**
- file path
 - configuration file **45**
 - virtual disk file **46**
- file system targets **55**
- file systems
 - NAS **28**
 - VMFS **56**
- finding available LUNs **29**
- fixed path selection policy **85**

G

- getting help for esxcli **84**
- getting preferred path **85**
- groups **48**
- GSX Server **60**

H

hard power operations **82**
 hardware iSCSI setup tasks **77**
 HBA properties **71**
 help for esxcli **84**
 host maintenance, vihostupdate35 **101**

I

inflating thin virtual disks **59**
 inheritance **75**
 initializing virtual disks **58**
 initiator alias **73**
 initiator iSCSI name **73**
 installing vMA **13**
 installing vSphere CLI
 Linux **10**
 Windows **12**
 iSCSI
 alias **73**
 authentication **71**
 disabling **76**
 initiator properties **73**
 LUNs **27**
 network portal information **72**
 network properties **73**
 parameter options **74**
 parameters, returning to default inheritance **75**
 path masking **96**
 setup examples **76**
 storage **99**
 ISO-8859-1 encoding **17**

K

Kerberos **33**

L

Linux
 executing commands **11**
 installing vSphere CLI **10**
 vSphere CLI **10**
 listing available LUNs **29**
 listing disk attributes **56**
 listing dynamic discovery targets **67**
 listing iSCSI initiator properties **73**
 listing iSCSI network portal information **72**
 listing iSCSI network properties **73**
 listing iSCSI parameter options **74**
 listing RDM attributes **61**
 listing software iSCSI loading status **75**
 listing static discovery targets **67**
 listing VMFS volume attributes **56**
 loading status, software iSCSI **75**
 logs **41**

LUNs

discovery **29**
 information **72**
 listing available **29**
 names **27**
 rescanning **30**
 VML names **99**

M

managing claim rules **92**
 managing diagnostic partitions **25**
 managing groups **48**
 managing NAS file systems **28**
 managing NMP **84**
 managing physical NICs **32**
 managing updates **42**
 managing users **48**
 managing virtual switches **36**
 managing VMkernel NICs **35**
 manipulating route entry **34**
 masking a path **96**
 Microsoft Windows Security Support Provider
 Interface **16**
 migrating virtual machines
 GSX Server **60**
 svmotion **44**
 vmkfstools **60**
 VMware Workstation **60**
 miscellaneous management commands **46**
 module options **47**
 multipathing **27, 99**
 mutual authentication, iSCSI **70**

N

naa.xxx device name **27**
 NAS datastores
 accessing **28**
 adding to ESX host **28**
 removing from ESX host **28**
 vicfg-nas **28**
 NetQueue configuration **47**
 network adapters
 duplex value **32**
 managing **32**
 speed **32**
 vicfg-nics **32**
 vicfg-vmknics **35**
 network portal information **72**
 networking commands **31**
 NIC binding, esxcli swiscsi nic **90**
 NICs **32**
 NMP **84, 85**
 NTP server specification **33**

O

options **17**
 order of precedence **14**

P

parameters
 command line **15**
 default inheritance (iSCSI) **75**
 partitions, diagnostic **25**
 path masking **96**
 path operations **86**
 path policies **85, 86, 87, 88**
 performance monitoring **39**
 Perl **9**
 physical compatibility mode RDM **57, 61**
 physical network adapters **32**
 physical network portal information **72**
 physical NICs **32**
 platform support **20**
 port groups **37**
 power operations
 hard **82**
 soft **82**
 powerop_mode values **82**
 precedence **14**
 preferred path **85, 86**
 PSA
 acronym **83**
 managing claim rules **93**
 PSP **87**
 PSP operations **86**

R

raw device mapping
 physical compatibility mode **61**
 virtual compatibility mode **61**
 raw disks
 cloning **59**
 RDM format **57**
 rdm format **57**
 RCLI See vSphere CLI
 RDMs
 attributes **61**
 disk format **57**
 physical compatibility mode **61**
 virtual compatibility mode **61**
 removing dynamic targets **68**
 removing static targets **68**
 renaming virtual disks **59**
 required parameters **13**
 rescanning adapters **30**
 rescanning LUNs **30**
 rescanning storage **30**

resxtp **19, 39**
 round robin operations **87**
 round robin path policy **87, 88**
 route manipulation **34**
 running commands
 Linux **10, 11**
 Windows **12**
 running commands from vMA **13**

S

SATP operations **88**
 scripts with vSphere CLI commands **18**
 session file for authentication **14**
 setting initiator alias **73**
 setting initiator iSCSI name **73**
 setting iSCSI network portal information **72**
 setting iSCSI network properties **73**
 setting iSCSI parameter options **74**
 setting preferred path **86**
 Shift_JIS encoding **17**
 Simple Network Management Protocol **40**
 snapshots **60**
 SNMP management **40**
 soft power operations **82**
 software iSCSI
 loading status **75**
 setup tasks **76**
 spanning partitions, vmkfstools **57**
 SSPI protocol **16**
 static targets, removing **68**
 storage
 commands **25**
 creating directories with vifs **51**
 rescanning **30**
 storage arrays, vicfg-mpath **27**
 storage management commands **25**
 supported authentication methods **69**
 supported disk formats **57**
 supported platforms **20**
 svmotion **44**
 syslog server specification **41**
 system logs **41**
 system management commands **39**

T

target information, vicfg-iscsi **71**
 target, for vmkfstools **55**
 thin virtual disks
 format **57**
 inflating **59**
 vmkfstools **59**
 third-party bundle **44**

U

- uninstalling vSphere CLI package
 - Linux **12**
 - Windows **13**
- unpacking vSphere CLI package **11**
- updates with vihostupdate **42**
- updating ESX Server 3i **101**
- updating VI 3.5 systems **101**
- uplink adapter **37**
- user management **48**
- using session files **14**

V

- vicfg-advcfg **46**
- vicfg-cfgbackup **39**
- vicfg-dns **31**
- vicfg-dumppart **25**
- vicfg-iscsi
 - authentication methods **69**
 - authentication options **69**
 - command syntax **63**
 - default inheritance for authentication **71**
 - default inheritance for parameters **75**
 - discovery target options **67**
 - information retrieval options **71**
 - initiator options **73**
 - iscsi parameter options **74**
 - network options **72**
 - options **64**
 - physical network portal options **72**
 - software iscsi options **75**
 - target information **71**
- vicfg-module
 - NetQueue configuration **47**
 - uses **47**
- vicfg-mpath **27**
- vicfg-mpath35 **99**
- vicfg-nas **28**
- vicfg-nics **32**
- vicfg-ntp **33**
- vicfg-rescan **30**
- vicfg-route **34**
- vicfg-scsidevs **29**
- vicfg-snmpp **40**
- vicfg-syslog **41**
- vicfg-user **48**
- vicfg-vmhbadevs **29**
- vicfg-vmknics **35**
- vicfg-volume **49**
- vicfg-vswitch **36**
- vifs **50**
- vihost parameter **18**

- vihostupdate
 - bulletins **42**
 - depots **42, 44**
 - third-party bundle **44**
 - update management **42**
- vihostupdate35 **101**
- virtual compatibility mode RDM **61**
- virtual disks
 - cloning **59**
 - creating **58**
 - deleting **59**
 - extending **60**
 - file path **46**
 - geometry **62**
 - inflating **59**
 - initializing **58**
 - options **57**
 - renaming **59**
- virtual machines
 - HBA names **99**
 - management commands **44**
 - managing **80**
 - migration with svmotion **44**
 - vmware-cmd **80**
- virtual switches
 - port groups **37**
 - vicfg-vswitch **36**
- vMA
 - environment variables **15**
 - installing **13**
 - multiple configuration files **15**
- VMFS
 - creating with vmkfstools **55, 56**
 - extending volumes **57**
 - listing attributes **56**
 - vmkfstools **53**
 - volume attributes **56**
- VMkernel **35, 47**
- VMkernel module options **47**
- VMkernel NICs **35**
- vmkfstools
 - command syntax **53**
 - command-specific options **54**
 - creating pass-through RDM **61**
 - creating virtual disk **58**
 - creating VMFS **55**
 - creating VMFS example **56**
 - deleting virtual disk **59**
 - device target **55**
 - disk formats **57**
 - disk partition target **55**
 - display disk geometry **62**

- extending virtual disk **60**
- extending volumes **57**
- file system options **55**
- file system targets **55**
- inflating thin virtual disk **59**
- initializing virtual disk **58**
- options **55**
- overview **53**
- RDM attributes **61**
- renaming virtual disk **59**
- snapshots **60**
- supported targets **55**
- syntax **53**
- target **55**
- virtual disk options **57**
- VMFS volume attributes **56**
- VML LUN names **27, 99**
- VMware Workstation **60**
- vmware-cmd
 - connection options **79**
 - examples **82**
 - general options **80**
 - server options **80**
 - virtual machine options **80**
- volume management **49**

- vSphere CLI
 - command-line **15**
 - configuration files **15**
 - environment variables **15**
 - execution options **17**
 - installing on Linux **10**
 - installing on Windows **12**
 - vicfg-iscsi **63**
- vSphere CLI package
 - downloading **11**
 - installing on Linux **10**
 - installing on Windows **12**
 - uninstalling **12, 13**
 - unpacking **11**
- vSphere Management Assistant **13**
- vSphere SDK for Perl **9**

W

- Windows
 - executing commands **12**
 - installing vSphere CLI **12**
 - using vSphere CLI **12**

Z

- zeroedthick disk format **57**