vSphere Command-Line Interface Installation and Scripting Guide

ESX 4.1

ESXi 4.1

vCenter Server 4.1

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-000274-00

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

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

Contents

About This Book

The *vSphere Command-Line Interface Installation and Scripting Guide*, explains how to install and use the VMware® vSphere Command-Line Interface (vCLI) and includes example scenarios and command overviews.

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

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.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to http://www.vmware.com/support/pubs.

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 Command-Line Interface Reference*, which is the companion document to this guide, is available on the vSphere CLI documentation page.

The vSphere SDK for Perl documentation explains how you can use the vSphere SDK for Perl and related utility applications to manage your vSphere environment. The documentation includes 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 vCLI and other prepackaged software. See "Deploying vMA" on page 17.

Background information for the tasks discussed in this manual is available in the vSphere documentation set. The vSphere documentation consists of the combined VMware vCenter Server and ESX/ESXi documentation and includes configuration guides, administrator's guides, guides for storage setup, and more.

Technical Support and Education Resources

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

1 Installing vCLI

You can install a vCLI package on a Linux or a Microsoft Windows system, or deploy the vSphere Management Assistant (vMA) on an ESX/ESXi host.

This chapter includes the following topics:

Installation Overview

The vCLI installer also installs vSphere SDK for Perl because vCLI commands run on top of the vSphere SDK for Perl. On Windows, the installation package includes vCLI, vSphere SDK for Perl, and prerequisite software. On Linux, the installation package includes vCLI and vSphere SDK for Perl. You are responsible for installing required prerequisite software.

- **vCLI packages.** You can install a vCLI package on a physical or virtual machine. See "Installing and Uninstalling vCLI on Linux" on page 11 and "Installing and Uninstalling vCLI on Windows" on page 15.

  After you have installed the package, which includes the vSphere SDK for Perl, you can run vCLI commands from the operating system command line. Each time you run a command, you specify the target server connection options directly or indirectly. See "Specifying Authentication Information" on page 20. You can also write scripts and manage your vSphere environment using those scripts.

- **vMA.** You can deploy vMA, a virtual machine that administrators can use to run scripts that manage vSphere, on an ESX/ESXi host. vMA includes vCLI and other prepackaged software in a Linux environment.

  vMA supports noninteractive login. If you establish an ESX/ESXi host as a target server, you can run vCLI commands against that server without additional authentication. If you establish a vCenter Server system as a target server, you can run most vCLI commands against all ESX/ESXi systems that server manages without additional authentication. See "Deploying vMA" on page 17.

Installing and Uninstalling vCLI on Linux

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

- Red Hat Enterprise Linux 5.2 (32 bit and 64 bit)
- SLES 10 (32 bit and 64 bit)
- SLES 11 (32 bit and 64 bit)
- Ubuntu 9.04 (32 bit and 64 bit)

If a version of vCLI or vSphere SDK for Perl is installed on your system, you must uninstall that version before you start the installation process.

IMPORTANT Install vCLI on Linux only if you are an experienced Linux administrator who knows how to use the system's package manager. Otherwise, use vMA. See "Deploying vMA" on page 17.

Installation Process

The vCLI package installer installs the vCLI scripts and the vSphere SDK for Perl. During installation, the installer checks whether prerequisites are installed. Depending on the type of prerequisite that is missing, the installer either stops the installation process or continues, as follows.

1 The installer checks whether the following required prerequisite packages are installed on the system:

OpenSSL	The vCLI requires SSL because most connections between the system on which you run the command and the target vSphere system are encrypted with SSL. The OpenSSL library (`libssl-dev` package) is not included in the default Linux distribution. Installation instructions for each platform are included below.
LibXML2	Used for XML parsing. The `libxml2` package is not included in the default Linux distribution. Installation instructions for each platform are included below.
e2fsprogs	Utilities for maintaining the ext2, ext3 and ext4 file systems. Required by the UUID Perl module.

2 If the software is found, the installer proceeds. Otherwise, the installer stops and informs you that you must install the software.

3 The installer checks whether the following recommended Perl modules are found, and whether the correct version is installed.

- Crypt-SSLeay-0.55 (0.55-0.9.7 or 0.55-0.9.8)
- IO-Compress-Base-2.005
- Compress-Zlib-2.005
- IO-Compress-Zlib-2.005
- Compress-Raw-Zlib-2.017
- Archive-Zip-1.26
- Data-Dumper-2.121
- XML-LibXML-1.63
- libwww-perl-5.805
- XML-LibXML-Common-0.13
- XML-NamespaceSupport-1.09
- XML-SAX-0.16
- Data-Dump-1.15
- URI-1.37
- UUID-0.03
- SOAP-Lite-0.710.08
- HTML-Parser-3.60
- version-0.78

4 If a recommended Perl module is not found at all, the installer installs it. If a different version of the module is found, the installer does not install it and proceeds with the installation process. At the end of the installation process, the installer informs you if the version on the system does not match the recommended version, and recommends that you install the version vCLI was tested with. You can install the modules using the package installer for your platform, the installation CD, or CPAN.

IMPORTANT The installer does not overwrite existing versions of recommended Perl modules. You must explicitly update those modules yourself.

If you have uninstalled a previous version of the vCLI or Remote CLI, and install vCLI 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 and shell (`setenv`, `export`, and so on). If you do not reset the PATH, the system might still go to the old location to find vCLI commands.

Installing Prerequisite Software for Red Hat Enterprise Linux 5.2

vCLI is supported on Red Hat Enterprise Linux 5.2, 32 bit and 64 bit.

Prerequisite software on RHEL includes required software and recommended Perl modules.

Required Software

If required software is not installed, the vCLI installer stops. You can install prerequisites using `yum`, the RHEL package installer (recommended), or from the installation DVD.

For example, if both OpenSSL development libraries and LibXML2 are missing, type the following at a command prompt:

```
yum install openssl-devel libxml2-dev
```

Recommended Perl Modules

When the installer finishes, it might issue a warning that the version of a module installed on your system does not match the version with which vCLI was tested. Install that version using `yum` or CPAN to resolve the issue. See "Installation Process" on page 12 for a complete list of modules.

After installing prerequisite software, you can install the vCLI itself. See "Installing the vCLI Package" on page 14.

Installing Prerequisite Software for SLES 10 and SLES 11

vCLI is supported on SLES 10, 32 bit and 64 bit and on SLES 11, 32 bit and 64 bit.

Prerequisite software on SLES includes required software and recommended Perl modules.

Required Software

If required software is not installed, the vCLI installer stops. You can install the prerequisite packages from the SLES 10 or SLES 11 SDK DVD. When you insert the DVD, it offers to auto run. Cancel the auto run dialog box and use `rpm` or the `yast` package installer to install OpenSSL or other missing required packages.

For example, if OpenSSL development libraries are missing, type the following at a command prompt:

```
yast -i openssl-devel
```

Some users might be authorized to use the Novell Customer Center and use `yast` to retrieve missing packages from there.

Recommended Perl Modules

When the installer finishes, it might issue a warning that the version of a module installed on your system does not match the version with which vCLI was tested. Install that version using `yast` or CPAN to resolve the issue. See "Installation Process" on page 12 for a complete list of modules.

After installing prerequisite software, you can install the vCLI itself. See "Installing the vCLI Package" on page 14.

Installing Prerequisite Software for Ubuntu Desktop 9.04

vCLI is supported on Ubuntu 9.04 32 bit and 64 bit.

Prerequisite software includes required software and recommended Perl modules.

Required Software

If required software is not installed, the vCLI installer stops. On Ubuntu you can use `apt` (advanced packaging tool) to keep a local repository of libraries up to date. You can use `apt` to install the required software.

To install required software on Ubuntu Desktop 9.04

1 Connect to the Internet.

2 Update the local repository of libraries from a terminal window.

```
sudo apt-get update
```

3 Install the required libraries from a terminal window. For Ubuntu Desktop 9.04 64-bit, you must install the 32-bit compatibility libraries or the `resxtop` and `esxcli` commands do not work.

32 bit
```
sudo apt-get install libssl-dev perl-doc liburi-perl libxml-libxml-perl
libcrypt-ssleay-perl
```

64 bit
```
sudo apt-get install libssl-dev perl-doc liburi-perl libxml-libxml-perl
libcrypt-ssleay-perl ia32-libs
```

Recommended Perl Modules

When the installer finishes, it might issue a warning that the version of a module installed on your system does not match the version with which vCLI was tested. Install that version using `apt-get` or CPAN to resolve the issue. See "Installation Process" on page 12 for a complete list of modules.

After installing prerequisite software, you can install the vCLI itself. See "Installing the vCLI Package" on page 14.

Installing the vCLI Package

Before you install version 4.1 of the vCLI, you must remove all previous versions of that software. The process differs from simply uninstalling vCLI.

To remove previous versions of vCLI

1 Run the uninstall script, for example, if you installed vCLI in the default location, run the following command:

```
/usr/bin/vmware-uninstall-vSphere-CLI.pl
```

2 Delete existing versions of `vSphere-CLI.xxxx.tar.gz` and delete the `vmware-vsphere-cli-distrib` directory.

To install vCLI on Linux

1 Untar the vCLI binary that you downloaded.

```
tar -zxvf VMware-vSphere-CLI-4.X.X-XXXXX.i386.tar.gz
```

A `vmware-vsphere-vcli-distrib` directory is created.

2 Log in as superuser and run the installer:

```
/<location>/sudo vmware-vsphere-cli-distrib/vmware-install.pl
```

3 To accept the license terms, type **yes** and press Enter.

4 Specify an installation directory, or press Enter to accept the default, which is `/usr/bin`.

A complete installation process has the following result:

- A success message appears.
- The installer lists different version numbers for required modules (if any).
- The prompt returns to the shell prompt.

If you accepted the defaults during installation, you can find the installed software in the following locations:

- **vCLI scripts** – `/usr/bin`
- **vSphere SDK for Perl utility applications** – `/usr/lib/vmware-vcli/apps`
- **vSphere SDK for Perl sample scripts** – `/usr/share/doc/vmware-vcli/samples`

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

After you install the vCLI, you can test the installation by running a command from the Windows command prompt.

To run a vCLI command on Linux

1 Open a command prompt.

2 Change to the directory where you installed the vCLI (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.

For example:

```
vicfg-nas --server my_esxserver --list
```

The system prompts you for a user name and password.

See Table 2-2, "vCLI Connection Options," on page 23 for a complete list of connection options.

Uninstalling the vCLI Package on Linux

You can use a script included in the installation to uninstall the vCLI package.

To uninstall the vCLI on Linux

1 Change to the directory where you installed the vCLI (default is `/usr/bin`).

2 Run the `vmware-uninstall-vSphere-CLI.pl` script.

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

Installing and Uninstalling vCLI on Windows

Before you can run vCLI commands on your Window system, you have to install the vCLI package and test the installation by running a command.

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

- Windows 2003 32 bit
- Windows XP SP3 32 bit
- Windows Vista Enterprise SP1 32 bit
- Windows 2008 64 bit

To install the vCLI Package on Windows

1 Download the vCLI Windows installer package.

You can find the installer on the VMware Communities page.

2 Start the installer.

3 (Optional) If prompted to remove older versions of vSphere SDK for Perl or vCLI, you can either accept or install the vCLI package on a different system.

IMPORTANT The installer replaces both the vSphere SDK for Perl and the vCLI. To keep an older version, install this package on a different system.

4 Click **Next** in the Welcome page.

5 To install the vCLI in a nondefault directory, click **Change** and select the directory.

The default location is `C:\Program Files\VMware\VMware vCLI.`

6 Click **Next**.

7 Click **Install** to proceed with the installation.

The installation might take several minutes to complete.

8 Reboot your system.

Without reboot, path settings might not be correct on your Windows platform.

After you install the vCLI and reboot your system, you can test the installation by running a command from the Windows command prompt.

To run a vCLI command on Windows

1 Open a command prompt.

2 Navigate to the directory in which the vCLI is installed.

```
cd C:\Program Files\VMware\VMware vSphere CLI\bin
```

3 Run the command, passing in connection options and other options.

On Windows, the extension `.pl` is required for most commands, but not for `esxcli`.

```
<command>.pl <conn_options> <params>
```

For example:

```
vicfg-nas.pl --server my_esxhost --list
```

The system prompts you for a user name and password.

See Table 2-2, "vCLI Connection Options," on page 23 for a complete list of connection options.

You can uninstall the vCLI package as you would other programs.

To uninstall the vCLI on Windows

1 Find the option for adding and removing programs on the Windows operating system you are using**.**

2 In the panel that appears, select **vSphere CLI**, and click **Remove**.

3 Click **Yes** when prompted.

The system uninstalls the vSphere SDK for Perl, the vCLI, and all prerequisite software.

Deploying vMA

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

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

1 Deploy vMA to an ESX/ESXi system that meets the hardware prerequisites.

   See the *vSphere Management Assistant Guide* for prerequisites and deployment details.

2 Configure vMA.

   When you boot vMA, you must specify the following required configuration information when prompted:

   - Network information (the default is often acceptable)
   - Host name for vMA.
   - Password for the vi-admin user. The vi-admin user has superuser privileges on vMA. You cannot log in to vMA as the root user.

3 (Optional) Add a vCenter Server system or one or more ESX/ESXi systems as targets. You can use the VMware vi-fastpass mechanism or Active Directory authentication, as explained in the *vSphere Management Assistant Guide*.

After you have specified a host as a vMA target, you can run vCLI commands against any ESX/ESXi target system without specifying connection options for that system explicitly. If you set up a vCenter Server system as a target server, you can connect any ESX/ESXi hosts that vCenter Server system manages using the `--vihost` option.

Running vCLI Commands 2

You can run vCLI commands from the command line and from scripts. Each command requires at a minimum the target server to run the command on. Users authorized to run commands on the target server do not have to specify authentication information. Other users must specify authentication information.

This chapter includes the following topics:

IMPORTANT If an ESXi system you target is in lockdown mode, you cannot run vCLI commands against that system directly. You must target a vCenter Server system that manages the ESXi system and use the `--vihost` option to specify the ESXi target. See "vCLI and Lockdown Mode" on page 23.

Overview of Running Commands

You can run vCLI commands interactively or in scripts in several ways.

- Open a command prompt on a Linux or Windows system on which you installed the vCLI. Enter commands into that command prompt.
- Access the vMA Linux console. Set up target servers and run vCLI commands against the targets without additional authentication.
- Prepare scripts that contain vCLI commands. Then run the scripts from a remote administration server that has the vCLI package installed or from the vMA Linux console. See "Using vSphere CLI Commands in Scripts" on page 25.

When you run commands against an ESX/ESXi host, you must be authenticated for that host. When you run commands against a vCenter Server system, and you are authenticated for that system, you can target all ESX/ESXi hosts that vCenter Server manages without additional authentication. See "Specifying Authentication Information" on page 20.

CAUTION If you specify passwords in plain text, you risk exposing the password to other users. The password might also become exposed in backup files. Do not provide plain-text passwords on production systems.

Follow one of the following approaches for protecting passwords.

- If you use a vCLI 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` script included in the `apps/session` directory. See "Using a Session File" on page 20.
- If you are running on a Windows system, you can use the `--passthroughauth` option. If the user who runs the command with that option is known, no password is required.

If you are running vMA, you can set up target servers and run most vCLI commands against target servers without additional authentication. See the *vSphere Management Assistant Guide*.

Specifying Authentication Information

vCLI allows you to run against multiple target servers from the same administration server. You must have the correct privileges to perform the actions on each target.

IMPORTANT vCLI 4.1 and later allows administrators to place ESXi hosts in lockdown mode for enhanced security. Only a vCLI or a vSphere Client connected to a vCenter Server system can make changes to ESXi hosts in lockdown mode. No users, not even the root user, can run vCLI commands against ESXi hosts in lockdown mode. See "vCLI and Lockdown Mode" on page 23 and the *Datacenter Administration Guide*.

Order of Precedence for vCLI Authentication

When you run a vCLI command, authentication happens in the order of precedence shown in Table 2-1. This order of precedence always applies. That means, for example, that you cannot override an environment variable setting in a configuration file.

Table 2-1. vCLI Authentication Precedence

Authentication	Description	See
Command line	Password (`--password`), session file (`--sessionfile`), or configuration file (`--config`) specified on the command line.	"Using a Session File" on page 20
Environment variable	Password specified in an environment variable.	"Using Environment Variables" on page 21
Configuration file	Password specified in a configuration file.	"Using a Configuration File" on page 21
Current account (Active Directory)	Current account information used to establish an SSPI connection. Available only on Windows.	"Using Microsoft Windows Security Support Provider Interface" on page 22
Credential store	Password retrieved from the credential store.	*vSphere Web Services SDK Programming Guide* and *vSphere SDK for Perl Programming Guide*.
Prompt the user for a password.	Password is not echoed to screen.	

Using a Session File

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

If you use a session file, 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/apps/session`

2 Run `save_session`.

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 --savesessionfile <location> --server <server>
```

For example:

Windows:
```
save_session.pl --savesessionfile C:\Temp\my_session --server my_server
--username <username> --password <password>
```

Linux:
```
save_session --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 When you run vCLI commands, pass in the session file using the `--sessionfile` option.

```
<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

On Linux, you can set environment variables in a Linux `bash` profile or on the command line by using a command like the following:

```
export VI_SERVER=<your_server_name_or_address>
```

On Windows, you can set environment variables in the Environment properties dialog box of the System control panel. For the current session, you can set environment variables at the command line by using a command like the following:

```
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 25 for an environment variable 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 2-2, "vCLI Connection Options," on page 23.

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

Pass in the configuration file when you run vCLI 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 = XX.XXX.XXX.XX
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.

Using Command-Line Options

You can pass in command-line options using option name and option value pairs in most cases. The following syntax results:

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

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

The following examples connect to the server as user `snow-white` with password `dwarf$`. The system displays help information for the command because the command is called with no options.

The first example (Linux) uses the backslash (\) 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

The `--passthroughauth` option, which is available if you run vCLI commands from a Microsoft Windows system, allows you to use the Microsoft Windows Security Support 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. No additional authentication is required if the user who runs the command is known by the computer from which you access the vCenter Server system and by the computer running the vCenter Server software.

If vCLI commands and the vCenter Server software run on the same computer, the user needs only a local account to run the command. If the vCLI command and the vCenter Server software run on different machines, the user who runs the command must have an account in a domain trusted by both machines.

SSPI supports several protocols. By default, it selects the `Negotiate` protocol, where client and server try 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 by using other options.

Example

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

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

vCLI and Lockdown Mode

Lockdown mode disables all direct root access to ESXi machines. You can only make changes to ESXi systems in lockdown mode by going through a vCenter Server system that manages the ESXi system. You can use the vSphere Client or vCLI commands that support the `--vihost` option. The following commands cannot run against vCenter Server systems and are therefore not available in lockdown mode:

- `vicfg-snmp`
- `vifs`
- `vicfg-user`
- `vicfg-cfgbackup`
- `vihostupdate`
- `vmkfstools`
- `esxcli`
- `vicfg-ipsec`

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

Common Options for vCLI Execution

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

IMPORTANT For connections, vCLI supports only the IPv4 protocol, not the IPv6 protocol. You can, however, configure IPv6 on the target host with several of the networking commands.

See "To run a vCLI command on Linux" on page 15 and "To run a vCLI command on Windows" on page 16 for usage examples.

Table 2-2. vCLI Connection Options

Option and Environment Variable	Description
`--config <cfg_file_full_path>` `VI_CONFIG=<cfg_file_full_path>`	Uses the configuration file at the specified location. Specify a path that is readable from the current directory.
`--credstore <credstore>`	Name of a credential store file. Defaults to `<HOME>/.vmware/credstore/vicredentials.xml` on Linux and `<APPDATA>/VMware/credstore/vicredentials.xml` on Windows. Commands for setting up the credential store are included in the vSphere SDK for Perl, which is installed with vCLI. The *vSphere SDK for Perl Programming Guide* explains how to manage the credential store.

Table 2-2. vCLI Connection Options (Continued)

Option and Environment Variable	Description
`--encoding <encoding>` `VI_ENCODING=<encoding>`	Specifies the encoding to be used. One of `cp936` (Simplified Chinese) `ISO-8859-1` (German), or `Shift_JIS` (Japanese). You can use `--encoding` to specify the encoding vCLI should map to when it is run on a foreign language system.
`--passthroughauth` `VI_PASSTHROUGHAUTH`	If you specify this option, the system uses the Microsoft Windows Security Support Provider Interface (SSPI) for authentication. Trusted users 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 vCLI on a Windows system and are connecting to a vCenter Server system.
`--passthroughauthpackage <package>` `VI_PASSTHROUGHAUTHPACKAGE=<package>`	Use this option with `--passthroughauth` to specify a domain-level authentication protocol to be used by Windows. By default, SSPI uses the `Negotiate` protocol, which means that client and server try 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 vCLI on a Windows system and connecting to a vCenter Server system.
`--password <passwd>` `VI_PASSWORD=<passwd>`	Uses the specified password (used with `--username`) to log in to the server. ■ If --server specifies a vCenter Server system, the user name and password apply to that server. If you can log in to the vCenter Server system, you need no additional authentication to run commands on the ESX/ESXi hosts that server manages. ■ If --server 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.
`--portnumber <number>` `VI_PORTNUMBER=<number>`	Uses the specified port to connect to the system specified by `--server`. Default is 443.
`--protocol <HTTP|HTTPS>` `VI_PROTOCOL=<HTTP|HTTPS>`	Uses the specified protocol to connect to the system specified by `--server`. Default is HTTPS.
`--savesessionfile <file>` `VI_SAVESESSIONFILE=<file>`	Saves a session to the specified file. The session expires if it has been unused for 30 minutes.
`--server <server>` `VI_SERVER=<server>`	Uses the specified ESX/ESXi or vCenter Server system. Default is `localhost`. If `--server` points to a vCenter Server system, you use the `--vihost` option to specify the ESX/ESXi host on which you want to run the command. A command is supported for vCenter Server if the `--vihost` option is defined.
`--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.

Table 2-2. vCLI Connection Options (Continued)

Option and Environment Variable	Description
`--username <u_name>` `VI_USERNAME=<u_name>`	Uses the specified user name. ■ If `--server` specifies a vCenter Server system, the user name and password apply to that server. If you can log in to the vCenter Server system, you need no additional authentication to run commands on the ESX/ESXi hosts that server manages. ■ If `--server` specifies an ESX/ESXi system, the user name and password apply to that system. 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.
`--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. **NOTE**: This option is not supported for each command. If supported, the option is included in the individual command option list.

Table 2-3 lists options not used as connection options that you can use when you run a vSphere CLI command.

Table 2-3. 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.

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`, `esx_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
```

Managing Hosts 3

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

The chapter includes the following topics:

Stopping, Rebooting and Examining Hosts with vicfg-hostops

You can shut down or reboot an ESX/ESXi host using the vSphere Client or the `vicfg-hostops` vCLI command. Shutting down a managed host disconnects it from the vCenter Server system, but does not remove the host from the inventory.

You can shut down a single host or all hosts in a datacenter or cluster.

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

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

  - If the host is not in maintenance mode, use `--force` to shut down the host and all running virtual machines.

    ```
    vicfg-hostops <conn_options> --operation shutdown --force
    ```

- **Multiple hosts** – To shut down all hosts in a cluster or datacenter, specify `--cluster` or `--datacenter`.

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

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

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

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

  - If the host is not in maintenance mode, use `--force` to shut down the host and all running virtual machines.

    ```
    vicfg-hostops <conn_options> --operation reboot --force
    ```

- **Multiple hosts** – You can specify `--cluster` or `--datacenter` to reboot all hosts in a cluster or datacenter.

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

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

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

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

Entering and Exiting Maintenance Mode with vicfg-hostops

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

If VMware DRS is in use, virtual machines that are running on a host that enters maintenance mode are migrated to another host automatically. If VMware DRS is not in use, `vicfg-hostops` suspends virtual machines. The host is in a state of Entering Maintenance Mode until all running virtual machines are suspended or migrated. When a host is entering maintenance mode, you cannot power on virtual machines on it or migrate virtual machines to it.

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

To enter maintenance mode

1 Run `vicfg-hostops --operation enter` to enter maintenance mode.

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

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

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

Backing Up Configuration Information with vicfg-cfgbackup

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

IMPORTANT The `vicfg-cfgbackup` command is available only for ESXi hosts. The command is not available for ESX hosts and is not available through a vCenter Server system connection.

Backup Tasks

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

The following tasks are required (See the *ESXi Installable and vCenter Server Setup Guide*):

1 Back up the configuration by using the `vicfg-cfgbackup` command.

2 Run the Recovery CD or repair operation

3 Restore the configuration by using the `vicfg-cfgbackup` command.

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

Backing Up Configuration Data

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

```
vicfg-cfgbackup <conn_options> -s /tmp/ESX_181842_backup.txt
```

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

Restoring Configuration Data

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

To restore ESXi configuration data

1 Power off all virtual machines that are running on the host that you want to restore.

2 Log in to a host on which vCLI is installed, or log in to vMA.

3 Run `vicfg-cfgbackup` with the `-l` flag to load the host configuration from the specified backup file.

- If you run the following command, you are prompted for confirmation.

  ```
  vicfg-cfgbackup <conn_options> -l /tmp/ESX_181842_backup.txt
  ```

- If you run the following command, you are not prompted for confirmation.

  ```
  vicfg-cfgbackup <conn_options> -l /tmp/ESX_181842_backup.txt -q
  ```

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

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

Using vicfg-cfgbackup from vMA

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

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

For example, backup for two ESXi 4.1 hosts (host1 and host2) with vMA deployed on both hosts works as follows:

- To back up a configuration (host 1 or host 2), run `vicfg-cfgbackup` from vMA running on either host1 or host2. Point to the host to back up using the `--server` command-line option.
- To restore the host1 configuration, run `vicfg-cfgbackup` from vMA running on host2 and point to host1 in the `--server` command-line option.
- To restore the host2 configuration, run on host1 and point to host2 in the `--server` command-line option.

Managing Host Updates 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 and later hosts. Run `vihostupdate35` against ESXi 3.5 hosts.

You cannot run `vihostupdate` against vCenter Server systems.

The `vihostupdate` command works with bulletins. A bulletin is a grouping of one or more VIBs (vSphere Installation Bundle). Each bulletin 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 VIB is included in the second bulletin.
- The VIB is obsoleted by another VIB in the second bulletin.

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

Bulletins are packaged as bundles or available in depots with associated `metadata.zip` files.

- A bundle is an archive that encapsulates VIBs and corresponding metadata in a self-contained depot that is useful for offline patching.If you use bundles, all patches and corresponding metadata are available as one ZIP file.
- You can also use a metadata.zip file that points directly to the location of the patch files in a depot.

The `vihostupdate` 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. See the *vSphere Command-Line Interface Reference* for information on all supported options. You can specify a patch by using a bundle ZIP file or the metadata ZIP file that describes the location 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. You can specify more than one bundle file at the command-line each time you run the command. Multiple bundles are usually required only if the update includes both a VMware bundle and a third-party bundle.

See the *ESXi Upgrade Guide*. For more information about installation, removal, and update of 3rd-Party Extensions in vSphere 4.x, see the *ESXi Setup Guide* and "Deploying Third-Party Bundles" on page 31.

To update a host using bundles

1 Power off all virtual machines running on the ESX/ESXi host by running the following command for each virtual machine.

```
vmware-cmd <conn_options> <vm-path> stop <powerop_mode>
```

Specify `hard` to force the power off operation or `soft` to have the system try to shut down the guest operating system.

2 Place the host into maintenance mode.

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

3 Check that the host is in maintenance mode. If necessary, shut down or migrate virtual machines.

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

4 Find out which bulletins are installed on the host.

```
vihostupdate <conn_options> --query
```

5 Find out which bulletins are available in the bundle.

```
vihostupdate <conn_options> --list --bundle http://<webserver>/rollup.zip
```

6 Find out which bulletins in the bundle are applicable to your host.

```
vihostupdate <conn_options> --scan --bundle http://<webserver>/rollup.zip
```

7 Install all or some bulletins from the bundle on the host. The following example installs both VMware bulletins and bulletins made available by a partner.

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

8 Verify that the bulletins are installed on your ESX/ESXi host.

```
vihostupdate <conn_options> --query
```

To update a host using depots

1 Power off all virtual machines running on the ESX/ESXi host by running the following command for each virtual machine.

```
vmware-cmd <conn_options> <vm-path> stop <powerop_mode>
```

Specify `hard` to force the power off operation or `soft` to have the system try to shut down the guest operating system.

2 Place the host into maintenance mode.

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

3 Check that the host is in maintenance mode. If necessary, shut down or migrate virtual machines.

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

4 List all bulletins in the depot given the `metadata.zip` file location.

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

5 Scan the depot for bulletins that are applicable to the host.

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

6 Install bulletins in the depot on the host.

- To install all bulletins, run the following command.

  ```
  vihostupdate <conn_options> --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 comma are not supported.

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

Deploying Third-Party Bundles

You can use the `--bundle` option to deploy a third-party bundle that you have downloaded on your Web server.

To deploy a third-party bundle

1 Power off all virtual machines that are running on the ESX/ESXi host.

```
vmware-cmd <conn_options> <vm-path> stop <powerop_mode>
```

Specify `hard` to force the power off operation or `soft` to have the system try to shut down the guest operating system.

2 Place the host into maintenance mode.

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

3 Check that the host is in maintenance mode. If necessary, shut down or migrate virtual machines.

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

4 Install the bundle.

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

Removing Bulletins from a Host

You can uninstall third-party bulletins or VMware extensions from your ESX/ESXi host.

IMPORTANT Do not remove bulletins that are VMware patches or updates.

To uninstall a bulletin

1 Power off all virtual machines that are running on the ESX/ESXi host.

```
vmware-cmd <conn_options> <vm-path> stop
```

2 Place the host into maintenance mode.

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

3 Check that the host is in maintenance mode, and shut down or migrate virtual machines if necessary.

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

4 Determine which bulletins are installed on your ESX/ESXi host.

```
vihostupdate <conn_options> --query
```

Note the bulletin ID for the bulletin to uninstall.

5 Run the `vihostupdate` command, specifying the bulletin to remove.

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

`vihostupdate` can remove only one bulletin at a time.

Managing VMkernel Modules with vicfg-module

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

To examine and set NetQueue VMkernel modules

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

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

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

```
vicfg-module <conn_options> --set-options 'intr_type_2 rx_ring_num=8' s2io
```

Configures a supported network interface to use NetQueue.

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

```
vicfg-module <conn_options> --get-options s2io
```

Verifies that the NetQueue module is configured.

Using vicfg-authconfig for Active Directory Configuration

`vicfg-authconfig` allows you to remotely configure Active Directory settings on ESX/ESXi hosts. Before you run the command on an ESX/ESXi host, you must prepare the host.

You can list supported and active authentication mechanisms, list the current domain and join or part from an Active Directory domain.

To prepare ESX/ESXi hosts for Active Directory Integration

1 Make sure the ESX/ESXi system and the Active Directory server are using the same time zone. The ESX/ESXi system's time zone is UTC by default.

2 Set up the ESX/ESXi system to have the same time as the Active Directory server.

3 Configure the ESX/ESXi system's DNS to be in the Active Directory domain.

You can now run `vicfg-authconfig` to add the host to the domain. A user who runs `vicfg-authconfig` to configure Active Directory settings must have the appropriate Active Directory permissions, and must have administrative privileges on the ESX/ESXi host.

To set up Active Directory

1 Install the ESX/ESXi host, as explained in the *Installation Guide*.

2 Install Windows Active Directory on a Windows Server running Windows 2000, Windows 2003, or Windows 2008.

3 Synchronize time between the ESX/ESXi system and Windows Active Directory (AD) and make sure the Windows AD Server and ESX/ESXi system are in the same time zone.

4 Test that the Windows AD Server can ping the ESX/ESXi host using the host name.

```
ping <ESX_hostname>
```

5 (Optional) If you cannot ping the ESX/ESXi system from the Windows AD Server, follow these steps to resolve the issue.

   a Append the following to `C:\Windows\System32\drivers\etc\hosts` file:

   ```
   <IP Address of ESX Server> <host name>
   ```

   b Make sure the `/etc/resolv.conf` file on the ESX/ESXi host contains the following string:

   ```
   nameserver <Windows AD IP Address>
   search <Domain Name of Windows AD>
   ```

6 Run the following vCLI command:

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

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

7 Check that a `Successfully Joined <Domain Name>` message appears.

8 Run `vicfg-authconfig --list` to verify the ESX/ESXi host is in the intended Windows AD domain.

See the *vSphere Command-Line Interface Reference* for a list of options and examples.

Managing Files 4

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

NOTE See "Managing Storage" on page 49 for information about storage manipulation commands.

This chapter includes the following topics:

Introduction to Virtual Machine File Management

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

NOTE Datastores are logical containers, analogous to file systems, that hide specifics of each storage device and provide a uniform model for storing virtual machine files. Datastores can be used for storing ISO images, virtual machine templates, and floppy images. The vSphere Client uses the term datastore exclusively. This manual will use the term datastore and VMFS (or NFS) volume to refer to the same logical container on the physical device.

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

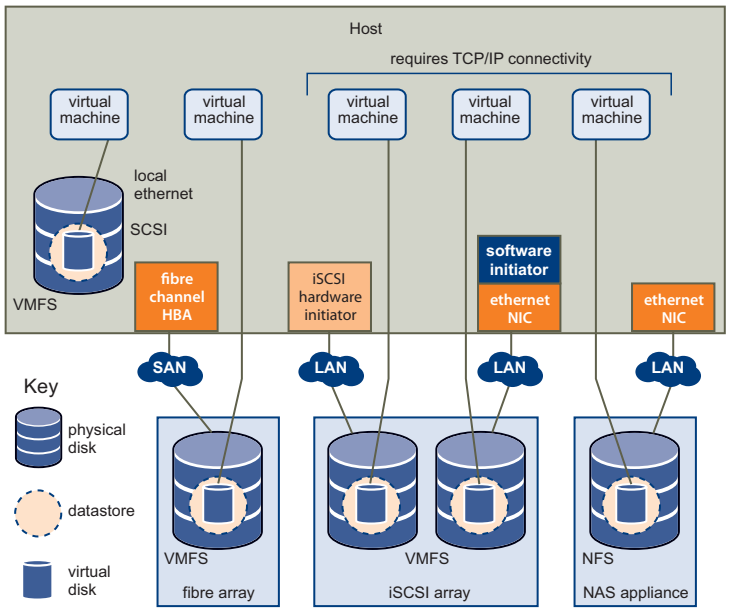
- Virtual Machine File System (VMFS) – High-performance file system that is optimized for storing virtual machines. Your host can deploy a VMFS datastore on any SCSI-based local or networked storage device, including Fibre Channel and iSCSI SAN equipment.

  As an alternative to using the VMFS datastore, your virtual machine can have direct access to raw devices and use a mapping file (RDM) as a proxy.

  You manage VMFS and RDMs with the vSphere Client or the `vmkfstools` utility.

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

Figure 4-1. Virtual Machines Accessing Different Types of Storage



Managing the Virtual Machine File System with vmkfstools

VMFS datastores primarily serve as repositories for virtual machines. You can store multiple virtual machines on the same VMFS volume. Each virtual machine, encapsulated in a set of files, occupies a separate single directory. For the operating system inside the virtual machine, VMFS preserves the internal file system semantics.

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

You use the `vmkfstools` vCLI 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. You can also use `vmkfstools` to set up and manage raw device mappings (RDMs).

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

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

vmkfstools Command Syntax

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

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

Option	Description	See
`<conn_options>`	Connection parameters. You must supply connection options. 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.	"vCLI Connection Options" on page 23.

Option	Description	See
<options>	One or more command-line options and associated values.	"Supported Command-Specific Options" on page 37.
<target>	Partition, device, or path to apply the command to.	"Supported vmkfstool Targets" on page 38.

Supported Command-Specific Options

You can use `vmkfstools` file system options to create or extend a VMFS file system, and to list file system attributes. Table 4-1 gives an overview.

Table 4-1. vmkfstools File System Option Overview

Option	Description	See
`--blocksize` `-b`	Uses the specified size for file system creation. Used with `--createfs`.	"Creating a VMFS File System" on page 38.
`--createfs` `-C`	Creates a VMFS file system.	"Creating a VMFS File System" on page 38.
`--queryfs` `-P`	Lists attributes of a file system.	"Listing VMFS Volume Attributes" on page 39.
`--setfsname` `-S`	Sets the label for the file system. Used with `--createfs`.	"Creating a VMFS File System" on page 38.
`--spanfs` `-Z`	Extends the VMFS file system.	"Extending VMFS Partitions by Spanning" on page 39.

You can use `vmkfstools` virtual disk options to create, clone, manipulate, and delete virtual disks and to manage RDMs (Raw Device Mappings). Table 4-2 gives an overview.

Table 4-2. vmkfstools Virtual Disk Option Overview

Option	Description	See
`--adaptertype` `-a`	Uses the specified type for disk creation. Used with `-c` and `-i`.	"Creating Virtual Disks" on page 41.
`--clonevirtualdisk` `-i`	Clones the specified virtual disk.	"Cloning Virtual or Raw Disks" on page 42.
`--createrdm` `-r`	Maps a raw disk to a file on a VMFS file system.	"Creating Virtual Compatibility Mode Raw Device Mappings" on page 45.
`--createrdmpassthru` `-z`	Maps a passthrough raw disk to a file on a VMFS file system.	"Creating Physical Compatibility Mode Raw Device Mappings" on page 45.
`--createvirtualdisk` `-c`	Creates a virtual disk.	"Creating Virtual Disks" on page 41.
`--deletevirtualdisk` `-U`	Deletes the specified virtual disk.	"Deleting Virtual Disks" on page 42.
`--diskformat` `-d`	Uses the specified format for disk creation. Used with `-c` and `-i`.	"Supported Disk Formats" on page 40.
`--extendvirtualdisk` `-X`	Extends the specified virtual disk.	"Extending Virtual Disks" on page 43.
`--geometry` `-g`	Displays virtual disk geometry.	"Displaying Virtual Disk Geometry" on page 43.
`--inflatedisk` `-j`	Converts a `thin` virtual disk to `eagerzeroedthick` format, preserving all existing data.	"Inflating Thin Virtual Disks" on page 41.

Table 4-2. vmkfstools Virtual Disk Option Overview (Continued)

Option	Description	See
`--renamevirtualdisk -E`	Renames the specified virtual disk.	"Renaming Virtual Disk" on page 42.
`--writezeros -w`	Cleans the virtual disk by writing zeros over all its data.	"Initializing Virtual Disks" on page 41.

Supported vmkfstool 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 pathname 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, a 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 pathname starting with `/vmfs/volumes`. For example, `/vmfs/volumes/datastore1/rh9.vmdk`

See "vmkfstools File System Options" on page 38.

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 40.

Device Target

Specify a device or logical volume using a pathname 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`	Local or SAN-based disks.
`/vmfs/devices/lvm`	ESX/ESXi logical volumes.
`/vmfs/devices/generic`	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 "vCLI Connection Options" on page 23.

Creating a VMFS File System

The `-C` option creates a VMFS file system on a specified partition, such as `naa.<naa_ID>:1`. The specified 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 When you run this command, you are not prompted 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.

The –C option has the following suboptions.

Option	Description
–b --blocksize	Block size for the VMFS-3 file system. The default file block size is 1MB. Valid block sizes for VMFS-3 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—vmkfstools interprets m or M to mean megabytes and k or K to mean kilobytes.
–S --setfsname	Volume label of a VMFS volume for the file system you are creating. Use –S only with the –C option. The label can be up to 128 characters long and cannot contain leading or trailing blank spaces. After you define a volume label, you can use it whenever you specify a VMFS volume in a call to vmkfstools. The volume label appears in listings generated for the Linux ls –l command and as a symbolic link to the VMFS volume under the /vmfs/volumes directory. You can change the VMFS volume label using the vSphere Client UI.

When you run vmkfstools –C, you can optionally specify the block size and volume label for the file system, and specify the location. For example:

- Create a VMFS-3 file system named my_vmfs with block size 2MB.

  ```
  vmkfstools <conn_options> –C vmfs3 –b 2m –S my_vmfs –vml.<vml_ID>:1
  ```

- Create a VMFS-3 file system named my_vmfs and use the default block size (1 MB).

  ```
  vmkfstools <conn_options> ––createfs vmfs3 ––setfsname my_n /vmfs/devices/disks/naa.<naa_id>:1
  ```

Listing VMFS Volume Attributes

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

```
–P ––queryfs
```

To list volume attributes

Run vmkfstools with connection options, the –P | ––queryfs option, and the volume to query.

```
vmkfstools <conn_options> ––queryfs /vmfs/volumes/my_vmfs
```

The listed attributes are version dependent but might include the VMFS version number, 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, as in the following example:

```
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 that is backing a VMFS file system goes offline, the number of extents and the available space change accordingly.

Extending VMFS Partitions by Spanning

You can run vmkfstools –Z to extend the VMFS file system with the specified head partition by spanning it across the partition specified by <span–partition>.

```
–Z | ––spanfs <span–partition> <head–partition>
```

The operation erases existing data on the spanned partition. A VMFS file system can have at most 32 partitions. The options does not work on VMFS-2 volumes. 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 specified in <span_partition>.

You can extend a partition by running `vmkfstools –Z` with connection options, the partition to span, and the head partition.

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

The command 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

A virtual disk is a file or set of files that appears as a physical disk drive to a guest operating system. Virtual disk files can be on the ESX/ESXi host or on a remote file system.

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

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 4-3 lists the supported formats.

Table 4-3. 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 the first time the virtual machine writes to disk. The virtual machine does not read stale data from disk.
`eagerzeroedthick`	Space required for the virtual disk is allocated at creation time. In contrast to `zeroedthick` format, the data remaining on the physical device is zeroed out during creation. It might take much longer to create disks in this format than to create other types of disks.
`thin`	Thin-provisioned virtual disk. Space required for the virtual disk is not allocated during creation, but is supplied, zeroed out, on demand at a later time. Convert thin-provisioned disks with the `–j` option. See "Inflating Thin Virtual Disks" on page 41.
`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 `thin`, with `vmkfstools`.

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`.

IMPORTANT If you run vCLI 4.x against vCLI 4.x systems, formats are supported as listed. Support is limited if you run vCLI 4.x against vCLI 3.5 systems.

Creating Virtual Disks

The –c option creates a virtual disk at the specified location on a VMFS volume.

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

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

You must specify the size of the virtual disk. Indicate the unit type for the `<size>` option by adding a suffix of `k` or `K` (kilobytes), `m` or `M` (megabytes), or `g` or `G` (gigabytes). If you do not specify a unit type, `vmkfstools` defaults to bytes.

You can specify the device driver and the disk format using `–a` and `–d`.

- `–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 the disk format, one of `zeroedthick`, `thin`, `eagerzeroedthick`. "Supported Disk Formats" on page 40 discusses all supported disk formats, including those supported by `createvirtualdisk`.

When you create a virtual disk, you can use the default file format and adapter or specify them explicitly.

- Run `vmkfstools –c` and specify the size and full path for the virtual disk. For example:

  ```
  vmkfstools <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 `my_vmfs`. This file represents an empty virtual disk that a virtual machine can access.

- Specify a file format and adapter for the virtual disk. You change the default adapter associated with the virtual machine (`buslogic`), using the `–a` option. Choices are `lsilogic` or `ide`.

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

  Creates a virtual disk associated with an lsilogic virtual adapter. See "Adding and Modifying VMkernel Network Interfaces with vicfg-vmknic" on page 117.

IMPORTANT The adapter type is set when you create the virtual disk. If you do not specify a type, the default (`buslogic`) is used. If you specify a type, it becomes the type for that disk. You cannot change the adapter type later.

Initializing Virtual Disks

The –w option erases a 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 that is 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.

You can run `vmkfstools` with the `--writezeros` option to initialize a virtual disk.

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

Inflating Thin Virtual Disks

A thin-provisioned virtual disk starts small and expands as more disk space is required. Administrators create a virtual disk in thin-provisioned format to save storage space.

The –j option converts a thin virtual disk to eagerzeroedthick format and preserves all existing data. vmkfstools allocates and zeroes out any blocks that were not allocated.

```
–j ––inflatedisk
```

You can inflate a virtual disk by running vmkfstools with connection options and the ––inflatedisk option.

```
vmkfstools <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 <conn_options> <location>
```

You can delete a virtual disk by running vmkfstools with connection options, the –U option, and the name of the virtual disk to be deleted.

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

Renaming Virtual Disk

The –E option renames a virtual disk file.

```
–E ––renamevirtualdisk <conn_options> <oldName> <newName>
```

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

You can rename a virtual disk by running vmkfstools –E and specify connection options, the old file path, and the new file path.

```
vmkfstools <conn_options> –E /vmfs/volumes/myvmfs/test.vmdk /vmfs/volumes/store/renamed.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 ––adaptertype <type>
```

The ––diskformat option specifies the disk format for the copy. diskformat defaults to zeroedthick if not specified. adaptertype defaults to buslogic if not specified. See "Supported Disk Formats" on page 40.

IMPORTANT To make a copy of the redo logs of an ESX/ESXi host while preserving their hierarchy, use the vifs –c command instead.

To clone a virtual or raw disk

1. Stop the source virtual machine.
2. Run vmkfstools –i, specifying the source and target and optional disk format and adapter type. For example:

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

   The clone process might take some time. The exact time depends on the size of your disk.
3. Configure a new virtual machine.
4. When asked to configure a disk, choose **Use an existing disk** and select the cloned disk you just created (myOS.vmdk in the example).
5. After the virtual machine is created, start it, log in, and change the IP address and host name.
6. After you have changed the IP address for the clone, you can power on the source virtual machine.

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.

To migrate Workstation virtual machines

1 Import a VMware Workstation or GSX Server disk into your `/vmfs/volumes/myVMFS/` directory or any subdirectory using `vmkfstools`.

2 Using the vSphere Client, create a new virtual machine using the **Custom** configuration option.

3 Select **Use an existing virtual disk** and attach 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] <v_disk>
```

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 that you add to the disk.

You specify the `newSize` option in kilobytes, megabytes, or gigabytes by adding a suffix of `k` or `K` (kilobytes), `m` or `M` (megabytes), or `g` or `G` (gigabytes). 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`. Extend virtual disks to `eagerzeroedthick` only if they are used for fault tolerance and clustering and have to be preallocated and zeroed. `-d` allows only `eagerzeroedthick`, it does not allow other disk formats.

To extend a virtual disk

1 Shut down the virtual machine associated with the disk, either by using the vSphere Client or by using `vmware-cmd` (see "Managing Virtual Machines" on page 89).

```
vmware-cmd -H <vc_system> --vihost <esx_host> /vmfs/volumes/Storage2/testvm/testvm.vmx stop
             <powerop_mode>
```

Specify `hard` to force the shutdown, or `soft` to have the system attempt to shut down the guest operating system.

2 Run `vmkfstools -X` with connection options, the new size, an optional new format, and the location for the virtual disk.

```
vmkfstools <conn_options> -X 50M /vmfs/volumes/Storage2/testvm/testvm.vmdk
```

Displaying Virtual Disk Geometry

The `-g` option retrieves 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 you import VMware Workstation virtual disks to ESX/ESXi host, you might see a disk geometry mismatch message. Geometry mismatch might also cause problems when you load a guest operating system or run a newly-created virtual machine.

Managing Raw Device Mapping Files

You can store virtual machine data directly on a SAN LUN instead of storing the data in a virtual disk file. Data on a SAN LUN is useful if you run applications in your virtual machines that must have information about the physical characteristics of the storage device. Mapping a SAN LUN also allows you to use existing SAN commands to manage storage for the disk.

When you map a LUN to a VMFS volume, the vCenter Server system creates a file called the RDM file, which points to the raw LUN. Encapsulating disk information in a file allows vCenter Server to lock the LUN so that only one virtual machine can write to it. As a result, all VMFS-3 file-locking mechanisms apply to RDMs.

IMPORTANT An RDM file has a `.vmdk` extension, but the file contains only disk information that describes the mapping to the LUN on the ESX/ESXi system. The actual data is stored on the LUN.

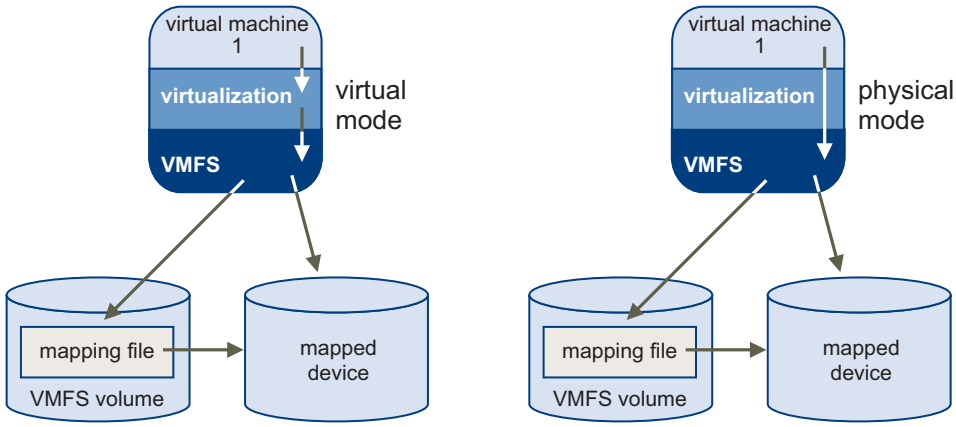
You cannot deploy a virtual machine from a template and store its data on a LUN. You can store its data only in a virtual disk file.

RDM Virtual and Physical Compatibility Modes

You can use RDMs in virtual compatibility or physical compatibility modes. Virtual mode specifies full virtualization of the mapped device. Physical mode specifies minimal SCSI virtualization of the mapped device. Physical mode allows the greatest flexibility for SAN management software.

- In virtual compatibility mode, the mapped device appears to the guest operating system exactly the same as a virtual disk file in a VMFS volume. The actual hardware characteristics are hidden. If you use a raw disk in virtual mode, you can realize the benefits of VMFS such as advanced file locking for data protection and snapshots for streamlining development processes. Virtual mode is also more portable across storage hardware than physical mode. A virtual mode mapped device presents the same behavior as a virtual disk file.
- In physical compatibility mode, the VMkernel passes all SCSI commands to the device, with one exception: the REPORT LUNs command is virtualized so that the VMkernel can isolate the LUN for the owning virtual machine. Otherwise, all physical characteristics of the underlying hardware are exposed. Physical mode is useful to run SAN management agents or other SCSI target-based software in the virtual machine. Physical mode also allows virtual-to-physical clustering for cost-effective high availability.

Figure 4-2. Physical and Virtual Compatibility Mode



Creating Virtual Compatibility Mode Raw Device Mappings

You can use the –r option to create a Raw Device Mapping (RDM) file in virtual compatibility mode on a VMFS-3 volume and to map a raw disk to this file.

```
–r ––createrdm <device> <rdm_file>
```

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

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

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.

To create and use a virtual compatibility mode RDM

1 Run vmkfstools –r with connection options, the device to map as a raw disk, and the name of the RDM file.

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

The command 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.

2 Configure a virtual machine to use the my_rdm.vmdk mapping file by using the vSphere Client or by adding the following lines to the virtual machine configuration file.

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

You can use vifs to copy the file from the remote ESX/ESXi system to your local machine and back. See "Using vifs to Manipulate Files on Remote ESX/ESXi Hosts" on page 46.

Creating Physical Compatibility Mode Raw Device Mappings

The –z option maps a physical compatibility mode raw device to a file on a VMFS volume.

```
–z ––createrdmpassthru <device> <map_file>
```

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>
```

To create a physical compatibility mode RDM

1 Run vmkfstools –z with connection options, the device to map as a raw disk, and the name of the RDM file.

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

The command creates a physical compatibility mode RDM file /vmfs/volumes/storage1/rdmpass.vmdk and maps the /vmfs/devices/disks/naa.<naa_id> raw disk to that file. You cannot use the name of a file that already exists.

2 Configure a virtual machine to use the rdmpass.vmdk mapping file by using the vSphere Client or by adding the following lines to the virtual machine configuration file:

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

Using vifs to Manipulate Files on Remote ESX/ESXi Hosts

In most cases, `vmkfstools` and other commands are used to manipulate virtual machine files. In some cases, you might need to view and manipulate files on remote ESX/ESXi hosts directly.

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

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

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

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

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

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

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

- **Host.** Host configuration files. You must specify the file's unique name identifier.

  Specify host locations using the `host/<path>` syntax.

- **Temp.** The `/tmp` directory and files in that directory.

  Specify temp locations using the `tmp/dir/subdir` syntax.

- **Datastores.** Datastore files and directories. You have two choices for specifying a datastore:

  - Datastore prefix style: `'[ds_name] relative_path'`. For example:

    ```
    '[myStorage1] testvms/VM1/VM1.vmx'(Linux) or "[myStorage1] testvms/VM1/VM1.vmx" (Windows)
    ```

  - URL style: `/folder/dir/subdir/file?dsName=<name>`. For example:

    ```
    '/folder/testvms/VM1/VM1.vmx?dsName=myStorage1' (Linux)
    "/folder/testvms/VM1/VM1.vmx?dsName=myStorage1" (Windows)
    ```

    The two example paths refer to a virtual machine configuration file for the virtual machine `VM1` in the `testvms/VM1` directory of the `myStorage1` datastore.

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

When you run `vifs`, you can specify the operation name and argument and one of the standard connection options discussed in Table 2-2. 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 available with `vifs`.

Options

`vifs` command-specific options allow you to retrieve and upload files from the remote host and perform a number of other operations. All `vifs` options work on datastore files or directories. Some options also work on host files and files in the `temp` directory. You must also specify connection options. See "vCLI Connection Options" on page 23.

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

You can list information about the remote directories in several ways.

- List the current datastores.

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

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

  ```
  osdc-cx700-02
  osdc-cx700-03
  osdc-cx700-02
  osdc-cx700-03
  osdc-cx700-04
  osdc-cx700-05
  ```

  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'
  ```

- List the contents of one of the datastores.

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

  The command lists the complete contents of the datastore.

- List the contents of one directory in the datastore.

  ```
  vifs <conn_options> --dir '[osdc-cx700-02] winxpPro-sp2'
  ```

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

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

The following example scenario illustrates other uses of `vifs`.

To manage files and directories on the remote ESX/ESXi system

1 Create a directory in the datastore.

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

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

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

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

3 Move a file into a virtual machine directory.

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

  A message indicates success or failure.

4 Retrieve one of the files from the remote ESX/ESXi system.

  The following example retrieves a log file for analysis.

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

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

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

Managing Storage 5

A virtual machine uses a virtual disk to store its operating system, program files, and other data associated with its activities. A virtual disk is a large physical file, or a set of files, that can be copied, moved, archived, and backed up.

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

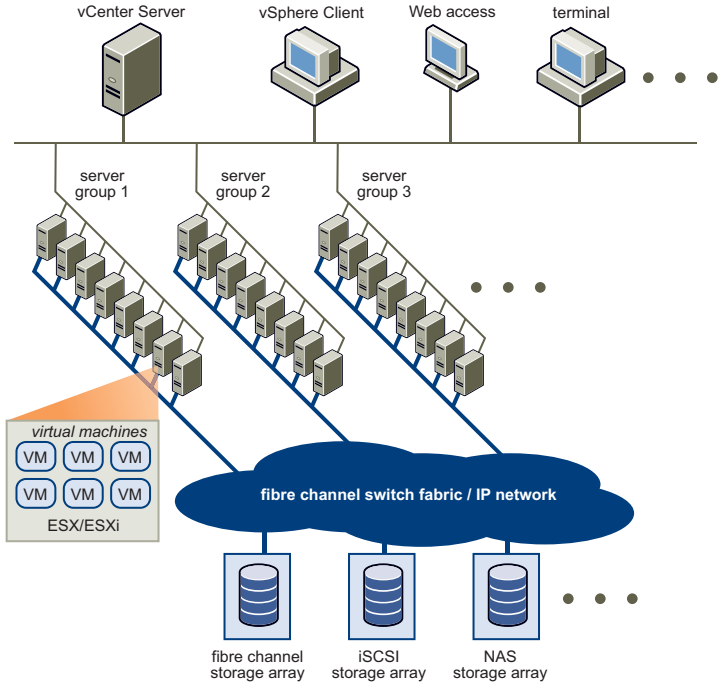
This chapter includes the following topics:

Chapter 6, "Managing iSCSI Storage," on page 65 discusses iSCSI storage management. Chapter 9, "Managing Third-Party Storage Arrays with esxcli," on page 97 explains how to manage the Pluggable Storage Architecture, including Path Selection Plugin (PSP) and Storage Array Type Plugin (SATP) configuration.

Introduction to Storage

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

Figure 5-1. vSphere Datacenter Physical Topology



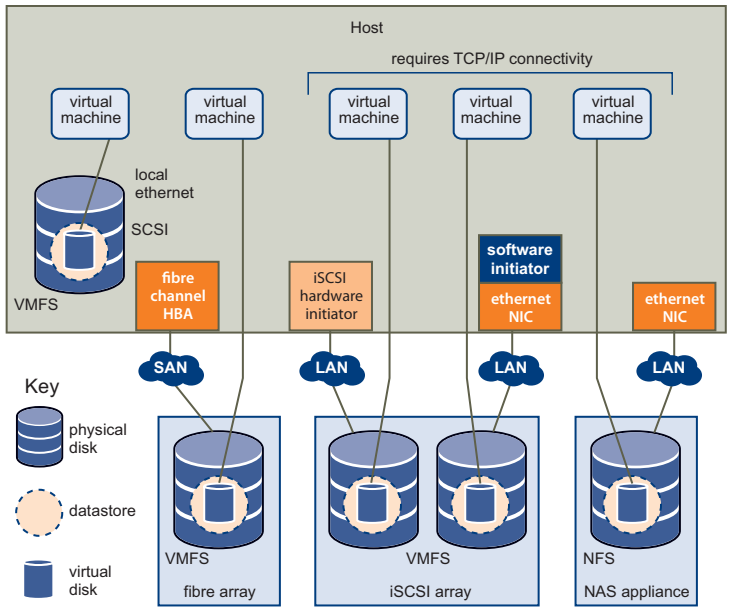
How Virtual Machines Access Storage

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

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

Figure 5-2 depicts five virtual machines using different types of storage to illustrate the differences between each type.

Figure 5-2. Virtual Machines Accessing Different Types of Storage



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

- **VMFS.** Use `vmkfstools` to create, modify, and manage VMFS virtual disks and raw device mappings. See "Managing the Virtual Machine File System with vmkfstools" on page 36.
- **Datastores.** Several commands allow you to manage datastores and are useful for multiple protocols.
  - **LUNs**. Use `vicfg-scsidevs` to display available LUNs and mappings for each VMFS volume to its corresponding partition. See "Examining LUNs with vicfg-scsidevs" on page 52.
  - **Path management.** Use `vicfg-mpath` to list information about Fibre Channel or iSCSI LUNs and to change a path's state. See "Managing Paths with vicfg-mpath" on page 53. Use the `esxcli` command to view and modify path policies. See "Managing Path Policies with esxcli" on page 55.
  - **Rescan.** Use `vicfg-rescan` to perform a rescan operation each time you reconfigure your storage setup. See "Rescanning Storage Adapters with vicfg-rescan" on page 63.
- **Storage devices.** Several commands manage only specific storage devices.
  - **NFS storage.** Use `vicfg-nas` to manage NAS storage devices. See "Managing NFS/NAS Datastores with vicfg-nas" on page 58.
  - **iSCSI storage.** Use `vicfg-iscsi` to manage both hardware and software iSCSI. See "Managing iSCSI Storage" on page 65.

Datastores

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

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

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

  As an alternative to using the VMFS datastore, your virtual machine can have direct access to raw devices and use a mapping file (RDM) as a proxy. See "Managing the Virtual Machine File System with vmkfstools" on page 36.
- **Network File System (NFS).** File system on a NAS storage device. ESX/ESXi supports NFS version 3 over TCP/IP. The host can access a designated NFS volume located on an NFS server, mount the volume, and use it for any storage needs.

Storage Device Naming

Each storage device, or LUN, is identified by several names.

- **Name.** A friendly name that the ESX/ESXi host assigns to a device based on the storage type and manufacturer, for example, DGC Fibre Channel Disk. This name is visible in the vSphere Client.
- **Device UID.** A universally unique identifier assigned to a device. The type of storage determines the algorithm used to create the identifier. The identifier is persistent across reboots and is the same for all hosts sharing the device. The format is often `naa.xxxxxxx` or `eui.xxxxxxxx`.
- **VML Name.** A legacy SCSI device name specific to VMware. Use the device UID instead.

The runtime name of the first path to the device is a path identifier and not a reliable identifier for the device. Runtime names are created by the host, and are not persistent. The runtime name has the format `vmhba#:C#:T#:L#`. You can view the runtime name using the vSphere Client.

Examining LUNs with vicfg-scsidevs

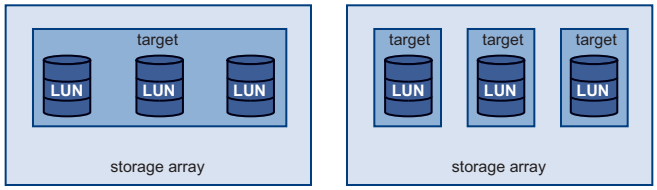
A LUN (Logical Unit Number) is an identifier for a disk volume in a storage array target.

Target and Device Representation

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

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

Figure 5-3. Target and LUN Representations



In Figure 5-3, three LUNs are available in each configuration. On the left, the host sees one target, but that target has three LUNs that can be used. Each LUN represents an individual storage volume. On the right, the host sees three different targets, each having one LUN.

Examining LUNs

Use `vicfg-scsidevs` to display information about available LUNs on ESX/ESXi 4.x hosts. For ESX/ESXi 3.5 systems, the corresponding command is `vicfg-vmhbadevs`.

IMPORTANT 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 and later.

You can run one of the following commands to examine LUNs.

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

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

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

  ```
  mpx.vmhba2:C0:T1:L0
      Device Type: cdrom
      Size: 0 MB
      Display Name: Local HL-DT-ST (mpx.vmhba2:C0:T1:L0)
      Plugin: NMP
      Console Device: /vmfs/devices/cdrom/mpx.vmhba2:C0:T1:L0
      Devfs Path: /vmfs/devices/cdrom/mpx.vmhba2:C0:T1:L0
      Vendor: SONY      Model: DVD-ROM GDRXX8XX Revis: 3.00
      SCSI Level: 5  Is Pseudo:   Status:
      Is RDM Capable:  Is Removable:
      Other Names:
          vml.000N000000XXXdXXXXXXXXaXXXaXX
          VAAI Status: nnnn

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

- List all logical devices with abbreviated information.

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

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

- List all device unique identifiers.

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

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

- List a specific logical device with its detailed information.

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

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

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

- Print HBA devices with identifying information.

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

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

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

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

Managing Paths with vicfg-mpath

To maintain a constant connection between an ESX/ESXi host and its storage, ESX/ESXi supports multipathing. Multipathing is a technique that lets you use more than one physical path for transferring data between the ESX/ESXi host and the external storage device.

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

The storage architecture in vSphere 4.0 and later supports a special VMkernel layer, Pluggable Storage Architecture (PSA). The PSA is an open modular framework that coordinates the simultaneous operation of multiple multipathing plugins (MPPs). You can manage PSA using `esxcli` commands. See "Managing Third-Party Storage Arrays with esxcli" on page 97. This section assumes you are using only PSA plugins included in vSphere by default.

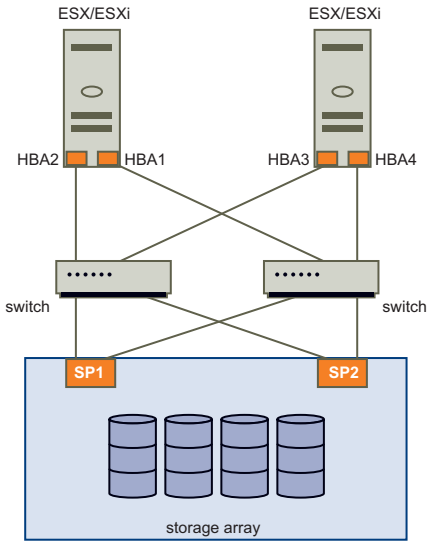
Multipathing with Local Storage and FC SANs

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

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

In Figure 5-4, multiple paths connect each host with the storage device. For example, if HBA1 or the link between HBA1 and the switch fails, HBA2 takes over and provides the connection between the server and the switch. The process of one HBA taking over for another is called HBA failover.

Figure 5-4. FC Multipathing



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

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

- List path information with `vicfg-mpath`. See "Listing Path Information" on page 54.
- Change path state with `vicfg-mpath`. See "Changing the State of a Path" on page 55.

  IMPORTANT Use `vicfg-mpath` for ESX/ESXi 4.0 or later. Use `vicfg-mpath35` for ESX/ESXi 3.5.

- Change path policies with `esxcli`. See "Setting Policy Details for Devices that Use Round Robin" on page 56.
- Mask paths with `esxcli`. See "Masking Paths with esxcli corestorage claimrule" on page 57.
- Rescan with `vicfg-rescan`. See "Rescanning Storage Adapters with vicfg-rescan" on page 63.

Listing Path Information

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

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

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

You can display information about paths by running `vicfg-mpath` with one of the following options:

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

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

- Display a short listing of all paths.

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

- List all paths with adapter and device mappings.

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

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

  ```
  vicfg-mpath <conn_options> --list
      -P sas.5001c231c79c4a00-sas.1221000001000000-naa.5000c5000289c61b
  ```

- List paths and detailed information by specifying the path runtime name.

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

  The return information includes the runtime name, device, device display name, adapter, adapter identifier, target identifier, plugin, state, transport, and adapter and target transport details.

- List detailed information for the paths for the device specified with `--device`.

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

Changing the State of a Path

You can temporarily disable paths for maintenance or other reasons, and enable the path when you need it again.

To disable a path

1 (Optional) List all devices and corresponding paths.

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

The display includes information about each path's state.

2 Set the state of a LUN path to off.

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

If you are changing a path's state:

- The change operation fails if I/O is active when the path setting is changed. Reissue the command.
- You must issue at least one I/O operation before the change takes effect.

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

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

Managing Path Policies with esxcli

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

Table 5-1. Supported Path Policies

Policy	Description
`VMW_PSP_FIXED`	The host always uses the preferred path to the disk when that path is available. If the host cannot access the disk through the preferred path, it tries the alternative paths. If you use the `VMW_PSP_FIXED` policy, use `esxcli nmp fixed` to set or get the preferred path
`VMW_PSP_FIXED_AP`	Extends the `VMW_PSP_FIXED` functionality to active-passive and ALUA mode arrays.
`VMW_PSP_MRU`	The host uses a path to the disk until the path becomes unavailable. When the path becomes unavailable, the host selects one of the alternative paths. The host does not revert back to the original path when that path becomes available again. There is no preferred path setting with the MRU policy. MRU is the default policy for active-passive storage devices and is required for those devices.
`VMW_PSP_RR`	The host uses an automatic path selection algorithm rotating through all available paths. This algorithm implements load balancing across all the available physical paths. Load balancing is the process of spreading server I/O requests across all available host paths. The goal is to optimize performance in terms of throughput (I/O per second, megabytes per second, or response times).

The type of array and the path policy determine the behavior of the host, as shown in Table 5-2.

Table 5-2. Path Policy Effects

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

To change the path policy

1 List all multipathing plugins loaded into the system.

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

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

2 Set the path policy using `esxcli`.

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

See Table 5-1.

3 (Optional) If you specified the `VMW_PSP_FIXED` policy, you must make sure the preferred path is set correctly.

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

```
esxcli <conn_options> nmp fixed getpreferred --device naa.xxx
```

b If necessary, change the preferred path.

```
esxcli <conn_options> nmp fixed setpreferred --device naa.xxx --path vmhba3:C0:T5:L3
```

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

Setting Policy Details for Devices that Use Round Robin

ESX/ESXi hosts can use multipathing for failover. With certain storage devices, ESX/ESXi hosts can also use multipathing for load balancing. To achieve better load balancing across paths, administrators can specify that the ESX/ESXi host should switch paths under certain circumstances. Different settable options determine when the ESX/ESXi host switches paths and what paths are chosen.

You can use `esxcli nmp roundrobin` to retrieve and set round robin path options on a device controlled by the `roundrobin` PSP.

IMPORTANT Only a limited number of storage arrays support round robin.

To view and manipulate round robin path selection settings

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

```
esxcli <conn_options> nmp roundrobin getconfig --device naa.xxx
```

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

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

  ```
  esxcli <conn_options> 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
  ```

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

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

Masking Paths with esxcli corestorage claimrule

With ESX/ESXi 4.0 and later, you use the MASK_PATH plugin instead of an advanced configuration option to mask paths.

IMPORTANT To convert ESX/ESXi 3.5 LUN masks to claim rule format, you might be able to use `esxcli corestorage claimrule convert`. See "Converting ESX 3.5 LUN Masks to Claim Rule Format" on page 107.

To mask paths

1 Run `esxcli corestorage claimrule list` to determine the next available rule ID.

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. Any third party multipathing plugins installed on your system use claim rules in this range.
- Rules 65436–65535 are reserved for internal use by VMware.

2 Decide which rule ID to use.

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.

3 Add the `MASK_PATH` plugin to the claim rule with the ID you decided to use.

```
esxcli <conn_options> corestorage claimrule add --plugin MASK_PATH --rule <ruleID>
    --type <type> -A <adapter>
```

4 Verify that the claim rule was added correctly.

```
esxcli <conn_options> corestorage claimrule list
```

5 Load the path claiming rules.

```
esxcli <conn_options> corestorage claimrule load
```

6 Release the device from the current plugin so that it can be claimed by another rule. For example, you might run the following command for each path:

```
esxcli <conn_options> corestorage claiming unclaim –t location –A vmhba0 –C 0 –T 0 –L 149
```

7 Run the path claiming rules, which include the newly added rules.

```
esxcli <conn_options> corestorage claimrule run
```

To unmask a path

1 Delete the `MASK_PATH` claim rule.

```
esxcli <conn_options> corestorage claimrule delete –r <rule#>
```

2 Verify that the claim rule was deleted correctly.

```
esxcli <conn_options> corestorage claimrule list
```

3 Reload the path claiming rules from the configuration file into the VMkernel.

```
esxcli <conn_options> corestorage claimrule load
```

4 Run `esxcli corestorage claiming unclaim` for each path to the masked device, for example:

```
esxcli <conn_options> corestorage claiming unclaim –t location –A vmhba0 –C 0 –T 0 –L 149
```

5 Run the path claiming rules.

```
esxcli <conn_options> corestorage claimrule run
```

Managing NFS/NAS Datastores with vicfg-nas

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

Capabilities Supported by NFS/NAS

ESX/ESXi hosts support the following shared storage capabilities on NFS volumes:

- VMware VMotion
- VMware DRS and VMware HA
- ISO images, which are presented as CD-ROMs to virtual machines
- Virtual machine snapshots

NAS stores virtual machine files on remote file servers that are accessed over a standard TCP/IP network. The NFS client built into the ESX/ESXi system uses NFS version 3 to communicate with NAS/NFS servers. For network connectivity, the host requires a standard network adapter.

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

To use NFS as a shared repository, you create a directory on the NFS server and then mount the directory as a datastore on all hosts. If you use the datastore for ISO images, you can connect the virtual machine's CD-ROM device to an ISO file on the datastore and install a guest operating system from the ISO file.

Adding and Deleting NAS File Systems

The following scenario illustrates how to list, add, and delete a NAS file system with `vicfg-nas`.

To manage a NAS file system

1 List all known NAS file systems.

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

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

```
No NAS datastore found
```

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

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

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

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

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

4 Delete one of the NAS file systems.

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

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

Migrating Virtual Machines with svmotion

The `svmotion` command moves a virtual machine's configuration file, and, optionally, its disks, while the virtual machine is running.

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

Storage VMotion Uses

Storage VMotion has several uses in administering your vSphere environment.

- Upgrade ESX/ESXi without virtual machine downtime. During an upgrade from ESX Server 2.x to ESX/ESXi 3.5 or later, you can migrate running virtual machines from a VMFS2 datastore to a VMFS3 datastore, and upgrade the VMFS2 datastore with no impact on virtual machines. You can then use Storage VMotion to migrate virtual machines back to the original datastore with no virtual machine downtime.
- Perform storage maintenance and reconfiguration. You can use Storage VMotion to move virtual machines off a storage device to allow maintenance or reconfiguration of the storage device without virtual machine downtime.
- Redistribute storage load. You can use Storage VMotion to manually redistribute virtual machines or virtual disks to different storage volumes to balance capacity or improve performance.

Storage VMotion Requirements and Limitations

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

- The virtual machine cannot have snapshots.
- Virtual machine disks must be in persistent mode or be raw device mappings (RDMs). For physical and virtual compatibility mode RDMs, you can migrate the mapping file only. For virtual compatibility mode RDMs, you can use the vSphere Client to convert to thick-provisioned or thin-provisioned disks during migration as long as the destination is not an NFS datastore. You cannot use the `svmotion` command to perform this conversion.
- The host on which the virtual machine is running must have a license that includes Storage VMotion.
- ESX/ESXi 3.5 hosts must be licensed and configured for VMotion. ESX/ESXi 4.0 and later hosts do not require VMotion configuration to perform migration with Storage VMotion.
- The host on which the virtual machine is running must have access to both the source and target datastores.
- A particular host can be involved in up to two migrations with VMotion or Storage VMotion at one time.
- vSphere supports a maximum of eight simultaneous VMotion, cloning, deployment, or Storage VMotion accesses to a single VMFS3 datastore, and a maximum of four simultaneous VMotion, cloning, deployment, or Storage VMotion accesses to a single NFS or VMFS2 datastore. A migration with VMotion involves one access to the datastore. A migration with Storage VMotion involves one access to the source datastore and one access to the destination datastore.

If you use the vSphere Client for migration with svmotion, the system performs a number of compatibility checks. These checks are not supported by the `svmotion` vCLI.

Running svmotion in Interactive Mode

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

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

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

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

Running svmotion in Noninteractive Mode

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

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

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

Square brackets indicate optional elements, not datastores.

The `--vm` option specifies the virtual machine and its destination. By default, all virtual disks are relocated to the same datastore as the virtual machine. This option requires the current virtual machine configuration file location. See "To determine the path to the virtual machine configuration file and disk file" on page 61.

The `--disks` option relocates individual virtual disks to different datastores. The `--disks` option requires the current virtual disk datastore path as an option. See "To determine the path to the virtual machine configuration file and disk file" on page 61.

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

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

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

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

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

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

To relocate a virtual machine's storage (including disks)

1 Determine the path to the virtual machine configuration file.

2 Run `svmotion`, for example:

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

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

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

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

2 Run `svmotion`, for example:

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

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

Managing Duplicate VMFS Datastores with vicfg-volume

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

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

When a LUN contains a VMFS datastore copy, you can mount the datastore with the existing signature or assign a new signature. The *ESX Configuration Guide* and the *ESXi Configuration Guide* discuss volume resignaturing in detail.

Mounting Datastores with Existing Signatures

You can mount a VMFS datastore copy without changing its signature if the original is not mounted. For example, you can maintain synchronized copies of virtual machines at a secondary site as part of a disaster recovery plan. In the event of a disaster at the primary site, you can mount the datastore copy and power on the virtual machines at the secondary site.

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

When you mount the VMFS datastore, ESX/ESXi allows both read and write operations to the datastore residing on the LUN copy. The LUN copy must be writable. The datastore mounts are persistent and valid across system reboots. Because ESX/ESXi prevents you from resignaturing the mounted datastore, unmount the datastore before resignaturing.

To mount a datastore

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

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

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

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

This command fails if the original copy is online.

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

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

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

Resignaturing VMFS Copies

Use datastore resignaturing to retain the data stored on the VMFS datastore copy. When resignaturing a VMFS copy, the ESX/ESXi host assigns a new UUID and a new label to the copy, and mounts the copy as a datastore distinct from the original.

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

When you perform datastore resignaturing, consider the following points:

- Datastore resignaturing is irreversible.
- The LUN copy that contains the VMFS datastore that you resignature is no longer treated as a LUN copy.
- A spanned datastore can be resignatured only if all its extents are online.
- The resignaturing process is crash and fault tolerant. If the process is interrupted, you can resume it later.
- You can mount the new VMFS datastore without a risk of its UUID conflicting with UUIDs of any other datastore, such as an ancestor or child in a hierarchy of LUN snapshots.

To resignature a VMFS copy

1 Make sure the copy is not mounted.

2 Run `vicfg-volume` with the `resignature` option.

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

The command returns to the prompt or signals an error.

After resignaturing, you might have to do the following:

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

Rescanning Storage Adapters with vicfg-rescan

You must perform a rescan operation each time you reconfigure your storage setup. You can rescan using the vSphere Client or the `vicfg-resan` vCLI command. The command requires that you specify the adapter for which you want to rescan the LUNs.

To rescan a storage adapter

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

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

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

Managing iSCSI Storage 6

ESX/ESXi systems include iSCSI technology to access remote storage using an IP network. You can use the vSphere Client or the `vicfg-iscsi` vCLI command to configure both hardware and software iSCSI storage for your ESX/ESXi system.

This chapter includes the following topics:

Chapter 5, "Managing Storage," on page 49 discusses other storage commands. Chapter 9, "Managing Third-Party Storage Arrays with esxcli," on page 97 explains how to manage the Pluggable Storage Architecture, including Path Selection Plugin (PSP) and Storage Array Type Plugin (SATP) configuration. See the *iSCSI SAN Configuration Guide* for a detailed discussion of iSCSI setup.

iSCSI Storage Overview

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

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

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

See the *iSCSI SAN Configuration Guide* for details on setup and failover scenarios.

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

Figure 6-1. iSCSI Storage

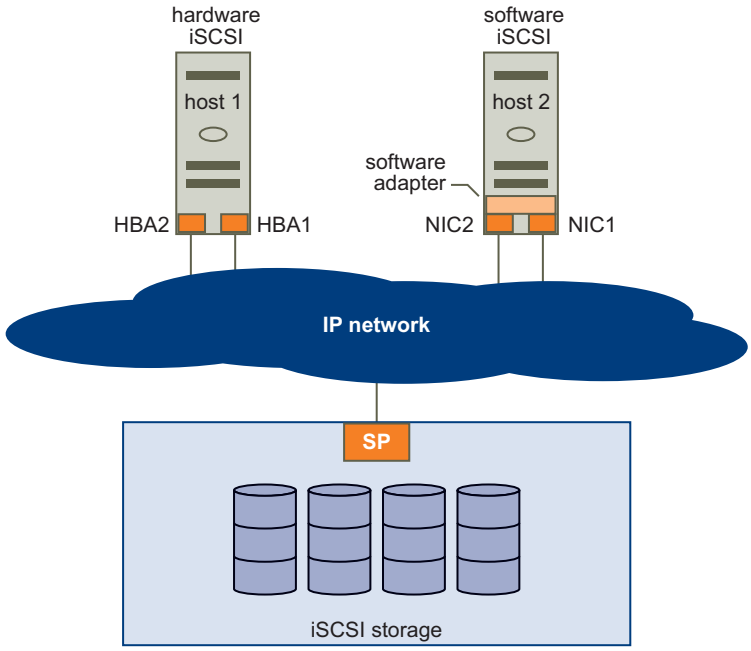


Figure 6-1 depicts to hosts that use different types of iSCSI initiators.

- The host on the left uses an independent hardware iSCSI adapter to connect to the iSCSI storage system.
- The host on the right uses a third-party Ethernet NIC with iSCSI offload capabilities.

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

Discovery Sessions

A discovery session is part of the iSCSI protocol. The discovery session returns the set of targets that you can access on an iSCSI storage system. 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 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 `vicfg-iscsi -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 you want to access. The iSCSI HBA must be in the same VLAN as both ports of the iSCSI array.

Discovery Target Names

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

- The IQN name uses the following format:

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

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

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

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

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

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

  For example:

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

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

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

Protecting an iSCSI SAN

When you plan your iSCSI configuration, take measures to improve the overall security of the iSCSI SAN. Your iSCSI configuration is only as secure as your IP network. By enforcing good security standards when you set up your network, you help safeguard your iSCSI storage.

Protecting Transmitted Data

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

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

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

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

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

Securing iSCSI Ports

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

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

Setting iSCSI CHAP

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

You can set up iSCSI authentication using the vSphere Client, as discussed in the *iSCSI SAN Configuration Guide* or using the `vicfg-iscsi` command, discussed in "Enabling iSCSI Authentication" on page 79. To use CHAP authentication, you must enable CHAP on both the initiator side and the storage system side. After authentication is enabled, it applies for targets to which no connection has been established, but does not apply to targets to which a connection is established. After the discovery address is set, the new volumes to which you add a connection are exposed and can be used.

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

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

Supported CHAP Levels

To set CHAP levels with `vicfg-iscsi`, specify one of the values in Table 6-1 for `<level>`. Only two levels are supported for independent hardware iSCSI.

Table 6-1. Supported Levels for CHAP

Level	Description	vSphere Client text	Supported
`chapProhibited`	Host does not use CHAP authentication. If authentication is enabled, specify `chapProhibited` to disable it.	Do not use CHAP	Software iSCSI Independent hardware iSCSI Dependent hardware iSCSI
`chapDiscouraged`	Host uses a non-CHAP connection, but allows a CHAP connection as fallback.	Do not use CHAP unless required by target	Software iSCSI Dependent hardware iSCSI
`chapPreferred`	Host uses CHAP if the CHAP connection succeeds, but uses non-CHAP connections as fallback.	Use CHAP unless prohibited by target	Software iSCSI Independent hardware iSCSI Dependent hardware iSCSI
`chapRequired`	Host requires successful CHAP authentication. The connection fails if CHAP negotiation fails.	Use CHAP	Software iSCSI Dependent hardware iSCSI

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

IMPORTANT Ensure that CHAP is set before you set mutual CHAP, and use compatible levels for CHAP and mutual CHAP.

Returning Authentication to Default Inheritance

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

- If you use the vSphere 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. Using this option resets both CHAP and mutual CHAP.

NOTE The `--reset-auth` option resets target-level CHAP authentication properties to be inherited from the adapter level. Resetting adapter-level properties is not supported.

iSCSI Storage Setup

You can set up iSCSI storage using the vSphere Client or the `vicfg-iscsi` command. After you have completed the initial setup (see "Setting Up Software iSCSI" on page 69 and "Setting Up Independent Hardware iSCSI" on page 72), you can examine and change options and parameters (see "Listing and Setting iSCSI Options" on page 77 and "Listing and Setting iSCSI Parameters" on page 77).

Setting Up Software iSCSI

Software iSCSI setup requires a number of high-level tasks. For each task, see the discussion of the corresponding command-line option in this chapter, the manpage (Linux), or the reference information.

1 Determine the HBA type and retrieve the HBA ID.

```
vicfg-iscsi --adapter --list
```

2 Enable software iSCSI for the HBA.

```
vicfg-iscsi --swiscsi --enable
```

3 (Optional) Check the status.

```
vicfg-iscsi --swiscsi --list
```

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

4 (Optional) Set the iSCSI name and alias.

```
vicfg-iscsi -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>
```

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

The two types of target differ as follows:

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

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

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

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

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

6 (Optional) Set the authentication information for CHAP (see "Setting iSCSI CHAP" on page 68 and "Returning Authentication to Default Inheritance" on page 69).

```
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>
```

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

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

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

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

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

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

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

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

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

Setting Up Dependent Hardware iSCSI

Dependent hardware iSCSI setup requires a number of high-level tasks. For each task, see the discussion of the corresponding command-line option in this chapter, the manpage (Linux), or the reference information.

1 Determine the HBA type and retrieve the HBA ID.

```
vicf-iscsi --adapter --list
```

2 (Optional) Set the iSCSI name and alias.

```
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>
```

3 Set up port binding by following these steps:

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

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

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

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

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

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

d Rescan the dependent hardware SCSI initiator.

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

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

The two types of target differ as follows:

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

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

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

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

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

5 (Optional) Set the authentication information for CHAP (see "Setting iSCSI CHAP" on page 68 and "Returning Authentication to Default Inheritance" on page 69).

```
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>
```

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

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

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

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

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

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

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

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

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

Setting Up Independent Hardware iSCSI

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

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

Hardware iSCSI setup requires a number of high-level tasks. For each task, see the discussion of the corresponding command-line option in this chapter, the manpage (Linux), or the reference information.

1 Determine the HBA type and retrieve the HBA ID.

```
vicf-iscsi --adapter --list
```

2 Configure the hardware initiator (HBA) by running `vicfg-iscsi -N` with one or more of the following options.

- `--list` – List network properties.
- `--ip <ip_addr>` – Set HBA IPv4 address.
- `--subnetmask <subnet_mask>` – Set HBA network mask.
- `--gateway <default_gateway>` – Set HBA gateway.
- `--set ARP=true|false` – Enable or disable ARP redirect.

You can also set the HBA IPv4 address and network mask and gateway in one command.

```
--ip <ip_addr> --subnetmask <subnet_mask> --gateway <default_gateway>
```

3 (Optional) Set the iSCSI name and alias.

```
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 or a static discovery address.

The two types of target differ as follows:

- With dynamic discovery, all storage targets associated with an IP address are discovered. Run the following command:

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

- With static discovery, you must specify the IP address and the iSCSI name of the storage target to be added. Run the following command:

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

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

5 (Optional) Set the authentication information for CHAP by running `vicfg-iscsi -A`.

You can set the information for per adapter, per discovery, and per target CHAP. See "Setting iSCSI CHAP" on page 68 and "Returning Authentication to Default Inheritance" on page 69.

```
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>
```

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

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

Mutual CHAP is not supported for independent hardware iSCSI storage.

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

7 After setup is complete, call `vicfg-rescan` to rescan all storage devices.

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>]
```

The following options are supported:

- `-D --discovery`
- `-S --static`
- `-A --authentication`
- `-P --phba`
- `-T --target`
- `-L --lun`
- `-N --network` (Independent hardware iSCSI only)
- `-p --pnp` (Independent 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.

Except for `--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* discuss 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` supports a comprehensive set of options, listed in Table 6-2.

Table 6-2. Options for vicfg-iscsi

Option	Suboptions	Description
`-A --authentication`		

	`-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>`
		Enables mutual authentication. You must enable authentication before you can enable mutual authentication.

`-A --authentication`		

	`-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>`
		Enables authentication using the specified options.

`-A --authentication`		
	`-l <adapter_name>` `--list <adapter_name>`	Lists supported 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.

Table 6-2. Options for vicfg-iscsi (Continued)

Option	Suboptions	Description
-L --lun		
	-l -t <target_ID> <adapter_name> --list --target_id <target_id> <adapter_name>	Lists LUN information for a specific target.
-N --network (Independent hardware iSCSI only)		
	-l <adapter_name> --list <adapter_name>	Lists network properties.
-N --network (Independent hardware iSCSI only)		
	-i <ip_addr> <adapter_name> --ip <ip_addr> <vmhba>	Sets the HBA IPv4 address to ip_addr.
-N --network (Independent hardware iSCSI only)		
	-s <subnet_mask> <adapter_name> --subnetmask <subnet_mask> <adapter_name>	Sets the HBA network mask to subnet_mask.
-N --network (Independent hardware iSCSI only)		
	-g <default_gateway> <adapter_name> --gateway <default_gateway> <adapter_name>	Sets the HBA gateway to default_gateway.
-N --network (Independent 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 (Independent hardware iSCSI only)		
	-l <adapter_name> --list <adapter_name>	Lists physical network portal options.
-p --pnp (Independent hardware iSCSI only)		
	-M <mtu_size> <adapter_name> --mtu <mtu-size> <adapter_name>	Sets physical network portal options.
-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.
-M --mtu		
	-p -M <mtu_size> <adapter_name> --pnp --mtu <mtu-size> <adapter_name>	Sets MTU size. Used with the --pnp option.
-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.

Table 6-2. Options for vicfg-iscsi (Continued)

Option	Suboptions	Description
–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.
–W --parameter		
	–W –o <param_name> –i <stor_ip_addr|stor_hostname> [:port_num>] [–n <iscsi_name>]] <adapter_name> –parameter ––reset <param_name> –ip <stor_ip_addr|stor_hostname> [:port_num>] [–name <iscsi_name>]] <adapter_name>	Returns parameters in discovery target or send target to default inheritance behavior.
–z ––reset_auth		
	–a –z –m <auth_method> –b [–i <stor_ip_addr|stor_hostname> [:<portnum>] [–n <iscsi_name>]] <adapter_name> ––authentication ––reset_auth ––method <auth_method> [––ip <stor_ip_addr|stor_hostname> [:<portnum>] [––name <iscsi_name>]] <adapter_name>	Resets target level authentication properties to be inherited from adapter level. Used with the ––authentication option.

Listing and Setting iSCSI Options

Use `vicfg-iscsi` information retrieval options to list external HBA properties, information about targets, and LUNs. You can use the following `vicfg-iscsi` options to list iSCSI parameters.

- Run `vicfg-iscsi -P|--phba` to list external (vendor-specific) properties of an iSCSI adapter.

  ```
  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.

- Run `vicfg-iscsi -T | --target` to list target information.

  ```
  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 "Discovery Target Names" on page 67.

- Run `vicfg-iscsi -L|--lun` to list LUN information.

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

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

- Run `vicfg-iscsi -L` with `-t` to list only LUNs on a specified target.

  ```
  vicfg-iscsi -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.

- Run `vicfg-iscsi -p|--pnp` to list physical network portal information for independent hardware iSCSI devices. You also use this option with `--mtu`.

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

  The system returns information about the MAC address, MTU, and current transfer rate.

- Run `vicfg-iscsi -I -l` to list information about the iSCSI initiator. ESX/ESXi systems use a software-based iSCSI initiator in the VMkernel to connect to storage. The command returns the iSCSI name, alias name, and alias settable bit for the initiator.

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

- Run `vicfg-iscsi -p -M` to set the MTU for the adapter. You specify the size and adapter name.

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

Listing and Setting iSCSI Parameters

You can list and set iSCSI parameters by running `vicfg-iscsi -W.` Table 6-3 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.

The parameters in Table 6-3 apply to software iSCSI and dependent hardware iSCSI.

Table 6-3. Settable iSCSI Parameters

Parameter	Description
`DataDigestType`	Increases data integrity. When data digest is enabled, the system performs a checksum over each PDUs data part and verifies using the CRC32C algorithm. **Note**: Systems that use Intel Nehalem processors offload the iSCSI digest calculations for software iSCSI, thus reducing the impact on performance. Valid values are `digestProhibited`, `digestDiscouraged`, `digestPreferred`, or `digestRequired`.
`HeaderDigest`	Increases data integrity. When header digest is enabled, the system performs a checksum over the header part of each iSCSI Protocol Data Unit (PDU) and verifies using the CRC32C algorithm.
`MaxOutstandingR2T`	Max Outstanding R2T defines the Ready to Transfer (R2T) PDUs that can be in transition before an acknowledgement PDU is received.
`FirstBurstLength`	Maximum amount of unsolicited data an iSCSI initiator can send to the target during the execution of a single SCSI command, in bytes.
`MaxBurstLength`	Maximum SCSI data payload in a Data-In or a solicited Data-Out iSCSI sequence, in bytes.
`MaxRecvDataSegLen`	Maximum data segment length, in bytes, that can be received in an iSCSI PDU.
`NoopInterval`	Time interval, in seconds, between NOP-Out requests sent from your iSCSI initiator to an iSCSI target. The NOP-Out requests serve as the ping mechanism to verify that a connection between the iSCSI initiator and the iSCSI target is active. Supported only at the initiator level.
`NoopTimeout`	Amount of time, in seconds, that can lapse before your host receives a NOP-In message. The message is sent by the iSCSI target in response to the NOP-Out request. When the `NoopTimeout` limit is exceeded, the initiator terminates the current session and starts a new one. Supported only at the initiator level.
`RecoveryTimeout`	Amount of time, in seconds, that can lapse while a session recovery is performed. If the timeout exceeds its limit, the iSCSI initiator terminates the session.
`DelayedAck`	Allows systems to delay acknowledgment of received data packets.

You can use the following `vicfg-iscsi` options to list parameter options.

- Run `vicfg-iscsi -W -l` to list parameter options for the HBA.

```
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>
```

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

Option	Result
`-i` and -n	Command applies to static targets.
Only -i	Command applies to the discovery address.
Neither `-i` nor -n	Command applies to per-adapter parameters.

- Run `vicfg-iscsi -W -l -k` to list iSCSI parameters and whether they are settable.

```
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>
```

- Run `vicfg-iscsi -W -j` to set iSCSI parameter options.

```
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>
```

  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.

If special characters are in the `<name>=<value>` sequence, for example, if you add a space, you must surround the sequence with double quotes ("`<name> = <value>`").

Returning Parameters to Default Inheritance

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

- If you use the vSphere Client to modify authentication settings, you deselect the **Inherit from Parent** check box before you can make a change to the discovery address or discovery target.
- If you use `vicfg-iscsi`, the value you set overrides the inherited value.

Inheritance is relevant only if you want to return a dynamic discovery address or a static discovery target to its inherited value. In that case, use the `--reset <param_name>` option, which requires the `--name` option for static discovery addresses, but not for dynamic discovery targets.

```
vicfg-iscsi <conn_options> --parameter --reset <param_name>
    --ip <stor_ip_addr | stor_hostname>[:port_num>] <adapter_name>
vicfg-iscsi <conn_options> -W -o <param_name>
    -i <stor_ip_addr|stor_hostname>[:port_num>] <adapter_name>
```

Enabling iSCSI Authentication

The `vicfg-iscsi -A -c` options enable iSCSI authentication. Mutual authentication is supported for software iSCSI and dependent hardware iSCSI, but not for independent hardware iSCSI. See "Setting iSCSI CHAP" on page 68.

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 the following requirements are met.

- CHAP authentication is already set up when you start setting up mutual CHAP.
- CHAP and mutual CHAP use different user names and passwords. The second user name and password are supported for mutual authentication on the storage side.
- CHAP and mutual CHAP use compatible CHAP levels.

Setting Up Ports for iSCSI Multipathing

With port binding, you create a separate VMkernel port for each physical NIC using 1:1 mapping. You can add all network adapter and VMkernel port pairs to a single vSwitch. The *iSCSI SAN Configuration Guide* explains how to specify port binding in detail.

To specify port binding

1 Find out which uplinks are available for use with iSCSI adapters.

```
esxcli <conn_options> swiscsi vmnic list –d <vmhba>
```

2 Connect the software iSCSI or dependent hardware iSCSI initiator to the iSCSI VMkernel ports by running the following command for each port.

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

3 Verify that the ports were added to the iSCSI initiator by running the following command:

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

4 Rescan the iSCSI initiator.

```
vicfg–rescan <conn_options> <vmhba>
```

5 (Optional) If there are active iSCSI sessions between your host and targets, discontinue them. See "Removing iSCSI Sessions" on page 81.

6 To disconnect the iSCSI initiator from the ports, run the following command.

```
esxcli <conn_options> swiscsi nic remove –n <port_name> –d <vmhba>
```

You can also use `esxcli swiscsi vmknic` to identify uplinks suitable for iSCSI and `esxcli swiscsi vmnic` to identify VMkernel network interfaces suitable for use with software iSCSI and dependent hardware iSCSI.

Managing iSCSI Sessions

You can use `esxcli swiscsi session` to list and manage iSCSI sessions for software iSCSI and dependent hardware iSCSI environments. The following sample scenario uses the available commands. Run `esxcli swiscsi session ––help` and each command with `––help` for reference information. The example uses a configuration file to log in to the host.

Listing iSCSI Sessions

- List a software iSCSI session at the adapter level.

  ```
  esxcli <conn_options> swiscsi session list –d <iscsi_adapter>
  ```

  For example:

  ```
  esxcli ––config /host–config–file swiscsi session list –d vmhba36
  ```

- List a software iSCSI session at the target level.

  ```
  esxcli <conn_options> swiscsi session list –t <iqn.xxxxx> –d <iscsi_adapter>
  ```

  For example:

  ```
  esxcli ––config /host–config–file swiscsi session list –t iqn.xxx –d vmhba36
  ```

Logging in to iSCSI Sessions

You can use `esxcli swiscsi session` to log in to a session.

- Log in to a session on the current software iSCSI or dependent hardware iSCSI configuration at the adapter level.

  ```
  esxcli <conn_options> swiscsi session add –d <iscsi_adapter>
  ```

  For example:

  ```
  esxcli ––config /host–config–file swiscsi session add –d vmhba36
  ```

- Log in to a session on the current software iSCSI or dependent hardware iSCSI configuration at the target level.

  ```
  esxcli <conn_options> swiscsi session add –t <iqn> –d <iscsi_adapter>
  ```

  For example:

  ```
  esxcli ––config /host–config–fileswiscsi session add –t iqn.xxx –d vmhba36
  ```

- Add duplicate sessions with target and session IDs in current software iSCSI or dependent hardware iSCSI configuration.

  ```
  esxcli <conn_options> swiscsi session add ––target <iqn.xxxx> ––isid <session_id>
      ––adapter <iscsi_adapter>
  ```

  `iqn.xxxx` is the target IQN, which you can determine by listing all sessions. `session_id` is the session's iSCSI ID. For example:

  ```
  esxcli ––config /host–config–file swiscsi session add –t iqn.xxx –s '00:02:3d:00:00:01'
      –d vmhba36
  ```

Removing iSCSI Sessions

You can use `esxcli swiscsi session` to remove iSCSI sessions.

- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration at the adapter level.

  ```
  esxcli <conn_options> swiscsi session remove –d <iscsi_adapter>
  ```

  For example:

  ```
  esxcli ––config /host–config–file swiscsi session remove –d vmhba36
  ```

- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration at the target level.

  ```
  esxcli <conn_options> swiscsi session remove –t <iqn> –d <iscsi_adapter>
  ```

  For example:

  ```
  esxcli ––config /host–config–file swiscsi session remove –t iqn.xxx –d vmhba38
  ```

- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration with target and session ID.

  ```
  esxcli <conn_options> swiscsi session remove ––target <iqn.xxxx> ––isid <session id>
      ––adapter <iscsi_adapter>
  ```

  `iqn.xxxx` is the target IQN, which you can determine by listing all sessions. `session_id` is the session's iSCSI ID.

  For example:

  ```
  esxcli ––config /host–config–file swiscsi session remove –t iqn.xxx –s '00:02:3d:01:00:01'
      –d vmhba36
  ```

Managing Users 7

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. You can use the vSphere Client or the vSphere SDK for all user management tasks. You can use the `vicfg-user` command to create, modify, delete, and list local direct access users and groups of users on an ESX/ESXi host. You cannot run this command against a vCenter Server system.

This chapter includes the following topics

Users and Groups in the vSphere Environment

Users, groups, and roles control who has access to your vSphere components and what actions each user can perform.

IMPORTANT You cannot use `vicfg-user` to create roles. You can use the roles that are predefined by the system.

vCenter Server and ESX/ESXi systems authenticate a user with a combination of user name, password, and permissions. The servers and hosts maintain lists of authorized users and the permissions assigned to each user.

Privileges define basic individual rights that are required to perform actions and retrieve information. ESX/ESXi and vCenter Server use sets of privileges, or roles, to control which users or groups can access particular vSphere objects. ESX/ESXi and vCenter Server provide a set of pre-established roles.

The privileges and roles assigned on an ESX/ESXi host are separate from the privileges and roles assigned on a vCenter Server system. When you manage a host using vCenter Server system, only the privileges and roles assigned through the vCenter Server are available. If you connect directly to the host by using the vSphere Client, only the privileges and roles assigned directly on the host are available.

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

vicfg-user Command Syntax

The `vicfg-user` syntax differs from other vCLI 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

`vicfg-user` command-specific options manipulate users and groups. You must also specify connection options. See "vCLI Connection Options" on page 23.

Option	Description
`--addgroup <group_list>` `-g <group_list>`	Comma-separated list of groups to add the user to.
`--adduser <user_list>` `-u <user_list>`	Adds the specified users to a specified group. Takes a comma-separated list of users.
`--entity <group|user>` `-e <group|user>`	Entity to perform the operation on. Specify either `user` or `group`.
`--group <name>` `-d <name>`	Group name of the group.
`--groupid <group_id>` `-D <group_id>`	Group ID of the group.
`--login <login_id>` `-l <login_id>`	Login ID of the user.
`--newpassword <p_wd>` `-p <p_wd>`	Password for the target user.
`--newuserid <UUID>` `-i <UUID>`	New UUID for the target user.
`--newusername <name>` `-n <name>`	New user name for the target user.
`--operation` `-o`	Operation to perform. Specify `add`, `modify`, `delete`, or `list`.
`--removegroup <group_list>` `-G <group_list>`	Comma-separated list of groups to remove the target user from.
`--removeuser <user_list>` `-U <user_list>`	Comma-separated list of users to be removed from the target group.
`--role <administrator|read-only|no-access>` `-r <administrator|read-only|no-access>`	Role for the target user or group. Specify one of `administrator`, `read-only`, or `no-access`. If you create a user without assigning permissions, the user has no permissions.
`--shell` `-s`	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 `yes` and `no`. This option is supported only against ESX. The option is not supported against ESXi.

Managing Users with vicfg-user

A user is an individual authorized to log in to an ESX/ESXi or vCenter Server system.

vSphere does not explicitly restrict users with the same authentication credentials from accessing and taking action within the vSphere environment simultaneously.

You manage users defined on the vCenter Server system and users defined on individual hosts separately.

- Manage ESX/ESXi defined users with the vSphere Client, the vSphere Web Services SDK, or `vicfg-user`.
- Manage vCenter Server users with the vSphere Client or the vSphere Web Services SDK.

Even if the user lists of a host and a vCenter Server system appear to have common users (for instance, a user called devuser), these users are separate users with the same name. The attributes of devuser in vCenter Server, including permissions, passwords, and so forth, are separate from the attributes of devuser on the ESX/ESXi host. If you log in to vCenter Server as devuser, you might have permission to view and delete files from a datastore. If you log in to an ESX/ESXi host as devuser, you might not have these permissions.

Users authorized to work directly on an ESX/ESXi host are added to the internal user list when ESX/ESXi is installed or can be added by a system administrator after installation. You can use `vicfg-user` to add users, remove users, change passwords, set group membership, and configure permissions.

CAUTION See the Authentication and User Management chapter of the *ESX Configuration Guide* or *ESXi Configuration Guide* for information about root users before you make any changes to the default users. Mistakes regarding root users can have serious access consequences.

Each ESX/ESXi host has a number of default users:

- The root user has full administrative privileges. Administrators use this login and its associated password to log in to a host through the vSphere Client. Root users can control all aspects of the host that they are logged on to. Root users can manipulate permissions, creating groups and users (on ESX/ESXi hosts only), working with events, and so on.
- The `vpxuser` user is a vCenter Server entity with root rights on the ESX/ESXi host, allowing it to manage activities for that host. The system creates `vpxuser` when an ESX/ESXi host is attached to vCenter Server. `vpxuser` is not present on the ESX/ESXi host unless the host is being managed through vCenter Server.
- Other users might be defined by the system, depending on the networking setup and other factors.

The following example scenario illustrates some of the tasks that you can perform.

To create, modify, and delete users

1 List the existing users.

```
vicfg-user <conn_options> -e user -o list
```

The list displays all users that are predefined by the system and all users that were added later.

2 Add a new user, specifying a login ID and password.

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

The command creates the user. By default, the command auto-generates a UID for the user and does not give shell access.

3 List the users again to verify that the new user was added and a UID was generated.

```
vicfg-user <conn_options> -e user -o list
USERS
-------------------
Principal -: root
Full Name -: root
UID -: 0
Shell Access -> 1
-------------------
...
--------------------
Principal -: user27
Full Name -:
UID -: 501
Shell Access -> 0
```

4 Modify the password for user `user27`.

```
vicfg-user <conn_options> -e user -o modify -l user27 -p 27_password2
```

The system might return `Updated user user27 successfully.`

5 Assign read-only privileges to the user (which currently has no access).

```
vicfg-user <conn_options> -e user -o modify -l user27 --role read-only
```

The system prompts whether you want to change the password, which might be advisable if the user does not currently have a password. Answer y or n. The system then updates the user.

```
Updated user user27 successfully.
Assigned the role read-only
```

6 List the existing groups.

```
vicfg-user <conn_options> -e group -o list
```

The system prints an extensive list of all groups and the users in each group.

7 Create a group.

```
vicfg-user <conn_options> -e group -o add -d test
```

The system adds the group, and assigns a group ID. When you now list all groups, the new group is included.

```
----------
Group Information:
Principal -: test
Full Name -:
GID -: 500
----------
```

8 Add user `user27` to the new group.

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

The system assigns the user to the group `test`. When you now list all groups, the new group and the assigned user are included.

```
----------
Group Information:
Principal -: test
Full Name -:
GID -: 500

Users in group test:
Principal -: user27
Full Name -:
-----------
```

9 Remove the user with login ID `user27`

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

The system removes the user and prints a message.

```
Removed the user user27 successfully.
```

Managing Groups with vicfg-user

You can efficiently manage some user attributes by creating groups. A group is a set of users that you manage through a common set of permissions.

A user can be a member of more than one group. When you assign permissions to a group, all users in the group inherit those permissions. Using groups can reduce the time it takes to set up your permissions model.

The group list in an ESX/ESXi host is drawn from a table that is maintained by the host. You can change the group list by using the vSphere Client or vCLI.

- Use the Users and Groups tab in the vSphere Client when the vSphere Client is connected directly to the host.
- Use the `vicfg-user` vCLI command.

Before you can add users to a group, you must create the group by using the `vicfg-user add` command, as in the following examples.

- Add `group40` to the existing groups. If you do not specify a group ID, the system assigns an ID for the group.

  ```
  vicfg-user <conn_options> -e group -o add -d group40 -D 55
  ```

- Create a group with predefined read-only privileges that you can later use to assign read-only privileges to multiple users.

  ```
  vicfg-user <conn_options> --entity group --operation add --group group42
  --groupid 4242 --role read-only
  ```

You can then add and remove users from the group, as in the following example scenario.

To add and remove users from groups

1 Add a user with user name `test` to a group `group42`.

  ```
  vicfg-user <conn_options> -e group -o modify -d group45 --adduser test
  ```

  You must specify the user name to add a user to a group. The user ID is not acceptable.

2 Add users with user names u1, u2, and u3 to group45, which has read-only privileges.

  ```
  vicfg-user <conn_options> -e group -o modify -d group42 --adduser u1,u2,u3
  ```

3 Remove the user with user name `u3` from the group.

  ```
  vicfg-user <conn_options> -e group -o modify -d group42 --removeuser u3
  ```

4 Remove the group with group name `group45`.

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

Managing Virtual Machines 8

You can manage virtual machines with the vSphere Client or the `vmware-cmd` vCLI command. Using `vmware-cmd` you can register and unregister virtual machines, retrieve virtual machine information, manage snapshots, turn the virtual machine on and off, add and remove virtual devices, and prompt for user input.

The chapter includes these topics:

Some virtual machine management utility applications are included in the vSphere SDK for Perl.

vmware-cmd Overview

`vmware-cmd` was included in earlier version of the ESX Service Console. A `vmware-cmd` vCLI has been available since ESXi version 3.0.

IMPORTANT Older service console versions of `vmware-cmd` support a set of connection options and general options that differ from the options in other vCLI commands. The `vmware-cmd` vCLI command supports these options. The vCLI command also supports the standard vCLI `--server`, `--username`, `--password`, and `--vihost` options. `vmware-cmd` does not support other connection options.

Connection Options for vmware-cmd

The `vmware-cmd` vCLI command supports only the following connection options. Other vCLI connection options are not supported, for example, you cannot use variables because the corresponding option is not supported.

Option	Description
`--server <host>` `-H <host>`	Target ESX/ESXi or vCenter Server system.
`--vihost <target>` `-h <target>`	When you run `vmware-cmd` with the `-H` option pointing to a vCenter Server system, use `--vihost` to specify the ESX/ESXi host to run the command against.
`-O <port>`	Alternative connection port. The default port number is 902.

Option	Description
`--username <username>` `–U <username>`	User who is authorized to log in to the host specified by `––server` or `––vihost`.
`--password <password>` `–P <password>`	Password for the user specified by `–U`.
`–Q <protocol>`	Protocol to use, either `http` or `https`. Default is `https`.

General Options for vmware-cmd

The `vmware–cmd` vCLI command supports the following general options.

Option	Description
`––help`	Prints a help message that lists 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.

Format for Specifying Virtual Machines

When you run `vmware–cmd`, the virtual machine path is usually required. You can specify the virtual machine using one of the following formats:

- Datastore prefix style: `'[ds_name] relative_path'`, for example:
  - `'[myStorage1] testvms/VM1/VM1.vmx'` (Linux)
  - `"[myStorage1] testvms/VM1/VM1.vmx"` (Windows)
- UUID-based path: `folder/subfolder/file`, for example:
  - `'/vmfs/volumes/mystorage/testvms/VM1/VM1.vmx'` (Linux)
  - `"/vmfs/volumes/mystorage/testvms/VM1/VM1.vmx"` (Windows)

Listing and Registering Virtual Machines

Registering or unregistering a virtual machine means adding the virtual machine to the vCenter Server or ESX/ESXi host inventory or removing the virtual machine.

IMPORTANT If you register a virtual machine with a vCenter Server system, and then remove it from the ESX/ESXi host, an orphaned virtual machine results. Call `vmware–cmd –s unregister` with the vCenter Server system as the target to resolve the issue.

The following example scenario lists all registered virtual machines on a vCenter Server, unregisters a virtual machine, and reregisters the virtual machine.

To list, unregister, and register virtual machines

1 Run `vmware–cmd –l` to list all registered virtual machines on a server.

```
vmware–cmd –H <vc_server> –U <login_user> –P <login_password> ––vihost <esx_host> –l
```

The command lists the VMX file for each virtual machine.

```
/vmfs/volumes/<storage>/winxpPro–sp2/winxpPro–sp2.vmx
/vmfs/volumes/<storage>/RHEL–lsi/RHEL–lsi.vmx
/vmfs/volumes/<storage>/VIMA0809/VIMA0809.vmx
.....
```

2 Run `vmware-cmd -s unregister` to remove a virtual machine from the inventory.

```
vmware-cmd -H <vc_server> -U <login_user> -P <login_password> --vihost <esx_host>
    -s unregister /vmfs/volumes/Storage2/testvm/testvm.vmx
```

The system returns `0` to indicate success, `1` to indicate failure.

NOTE When you run against a vCenter Server system, you must specify the datacenter and the resource pool to register the virtual machine in. The default datacenter is `ha-datacenter` and the default resource pool is `Resources`.

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.

---

3 Run `vmware-cmd -l` again to verify that the virtual machine was removed from the inventory.

4 Run `vmware-cmd -s register` to add the virtual machine back to the inventory.

```
vmware-cmd -H <vc_server> -U <login_user -P <login_password --vihost <esx_host> -s register
              /vmfs/volumes/Storage2/testvm/testvm.vmx
```

The system returns `0` to indicate success, `1` to indicate failure.

Retrieving Virtual Machine Attributes

`vmware-cmd` includes a number of options for retrieving information about a virtual machine. Each option requires that you specify the virtual machine path (see "Format for Specifying Virtual Machines" on page 90). You must also specify connection options, which differ from other vCLI commands (see "Connection Options for vmware-cmd" on page 89).

You can use `vmware-cmd` options to retrieve a number of different virtual machine attributes.

- The `getuptime` option retrieves the uptime of the guest operating system on the virtual machine, in seconds.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
              /vmfs/volumes/Storage2/testvm/testvm.vmx getuptime

getuptime() = 17921
```

- The `getproductinfo product` option lists the VMware product the virtual machine runs on.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
              /vmfs/volumes/Storage2/testvm/testvm.vmx getproductinfo product
```

The return value is `esx` (VMware ESX), `embeddedESX` (VMware ESXi), or `unknown`.

- The `getproductinfo platform` option lists the platform the virtual machine runs on.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
              /vmfs/volumes/Storage2/testvm/testvm.vmx getproductinfo platform
```

The return value is `win32-x86` (x86-based Windows system, `linux-x86` (x86-based Linux system), or `vmnix-x86` (x86-based ESX/ESXi microkernel).

- The `getproductinfo build`, `getproductinfo majorversion`, or `getproductinfo minorversion` options retrieve version information.
- The `getstate` option retrieves the execution state of the virtual machine, which can be `on`, `off`, `suspended`, or `unknown`.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
              /vmfs/volumes/Storage2/testvm/testvm.vmx getstate
getstate() = on
```

- The `gettoolslastactive` option indicates whether VMware Tools is installed and whether the guest operating system is responding normally.

  ```
  vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
                /vmfs/volumes/Storage2/testvm/testvm.vmx gettoolslastactive
  ```

  The command returns an integer indicating how much time has passed, in seconds, since the last heartbeat was detected from the VMware Tools service. This value is initialized to zero when a virtual machine powers on. The value stays at zero until the first heartbeat is detected. After the first heartbeat, the value is always greater than zero until the virtual machine is power cycled again. The command returns one of the following:

  - 0 – VMware Tools is not installed or not running.
  - 1 – Guest operating system is responding normally.
  - 5 – Intermittent heartbeat. There might be a problem with the guest operating system.
  - 100 – No heartbeat. Guest operating system might have stopped responding.

NOTE You usually use the `vmware-cmd guestinfo` option only when VMware Support instructs you to do so. The command is therefore not discussed in this document.

Managing Snapshots with vmware-cmd

A snapshot captures the entire state of the virtual machine at the time you take the snapshot.

Virtual machine state includes the following aspects of the virtual machine.

- Memory state – Contents of the virtual machine's memory.
- Settings state – Virtual machine settings.
- Disk state – State of all the virtual machine's virtual disks.

When you revert to a snapshot, you return these items to the state they were in at the time you took the snapshot. If you want the virtual machine to be running or to be shut down when you start it, make sure that it is in that state when you take the snapshot.

You can use snapshots as restoration points when you install update packages, or during a branching process, such as installing different versions of a program. Taking snapshots ensures that each installation begins from an identical baseline. The *Virtual Machine Administration* manual discusses snapshots in detail.

IMPORTANT Use the vSphere Client to revert to a named snapshot. `vmware-cmd` only supports reverting to the current snapshot.

Taking Snapshots

You can take a snapshot while a virtual machine is running, shut down, or suspended. If you are in the process of suspending a virtual machine, wait until the suspend operation has finished before taking a snapshot.

If a virtual machine has multiple disks in different disk modes, you must shut down the virtual machine before taking a snapshot. For example, if you have a special-purpose configuration that requires you to use an independent disk, you must shut down the virtual machine before taking a snapshot.

To take a snapshot

1 (Optional) If the virtual machine has multiple disks in different disk modes, shut down the virtual machine.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
              /vmfs/volumes/Storage2/testvm/testvm.vmx stop soft
```

2 (Optional) Check that the shut down operation has been completed.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
              /vmfs/volumes/Storage2/testvm/testvm.vmx getstate
```

3 Run vmware-cmd with the createsnapshot option.

You must specify the description, quiesce flag (0 or 1) and memory flag (0 or 1).

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
           /vmfs/volumes/Storage2/testvm/testvm.vmx createsnapshot VM1Aug09
           'test snapshot August 09' 0 0
```

4 Check that the virtual machine has a snapshot using the hassnapshot option.

The call returns 1 if the virtual machine has a snapshot and returns 0 otherwise.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
              /vmfs/volumes/Storage2/testvm/testvm.vmx hassnapshot

hassnapshot () = 1
```

Reverting and Removing Snapshots

You can use vmware-cmd to revert to the current snapshot or to remove a snapshot.

IMPORTANT You cannot use vmware-cmd to revert to a named snapshot. Use the vSphere Client to revert to a named snapshot.

Run vmware-cmd with the revertsnapshot option to revert to the current snapshot. If no snapshot exists, the command does nothing and leaves the virtual machine state unchanged.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
             /vmfs/volumes/Storage2/testvm/testvm.vmx revertsnapshot
```

Run vmware-cmd with the removesnapshots option to remove all snapshots associated with a virtual machine. If no snapshot exists, the command does nothing.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
             /vmfs/volumes/Storage2/testvm/testvm.vmx removesnapshots
```

Powering Virtual Machines On and Off

You can start, reboot, stop, and suspend virtual machines using vmware-cmd. You must supply a value for the powerop_mode flag, which can be soft or hard.

IMPORTANT You must have the current version of VMware Tools installed and running in the guest operating system to use a soft power operation.

- **Soft power operations**. When you specify soft as the powerop_mode value, the result of the call depends on the operation.

Operation	Result
Stop	vmware-cmd attempts to shut down the guest operating system, and then powers off the virtual machine.
Reset	vmware-cmd attempts to shut down the guest operating system, and then reboots the virtual machine.
Suspend	vmware-cmd attempts to run a script in the guest operating system before suspending the virtual machine.

- **Hard power operations**. vmware-cmd immediately and unconditionally shuts down, resets, or suspends the virtual machine.

The following examples illustrate how to use `vmware-cmd`.

- **Start** – Use the `start` option to power on a virtual machine or to resume a suspended virtual machine. The powerop_mode, either `hard` or `soft`, is required.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
            /vmfs/volumes/Storage2/testvm/testvm.vmx start soft
```

- **Reset** – When you reset the virtual machine with the `soft` power_op mode (the default), the guest operating system is shut down before the reset.

  a Check that VMware tools is installed so that you can reset the virtual machine with the default `power_op` mode, which is `soft`.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
            /vmfs/volumes/Storage2/testvm/testvm.vmx gettoolslastactive
```

  See "Retrieving Virtual Machine Attributes" on page 91.

  b Use the `reset` option to shut down and restart the virtual machine.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
            /vmfs/volumes/Storage2/testvm/testvm.vmx reset soft
```

  If VMware Tools is not currently installed on the virtual machine, you can perform only a hard reset operation.

- **Suspend** – You have two options for suspending a virtual machine.
  - The `suspend` option with the `hard` powerop mode unconditionally shuts down a virtual machine.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
            /vmfs/volumes/Storage2/testvm/testvm.vmx suspend hard
```

  - The `suspend` option with the `soft` powerop mode runs scripts that result in a graceful shut-down of the guest operating system and shuts down the virtual machine. VMware Tools must be installed for `soft` powerop_mode.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
            /vmfs/volumes/Storage2/testvm/testvm.vmx suspend soft
```

Connecting and Disconnecting Virtual Devices

You can add and remove virtual devices by using the `connectdevice` and `disconnectdevice` options. The selected guest operating system determines which of the available devices you can add to a given virtual machine.

The virtual hardware that you add appears in the hardware list that is displayed in the Virtual Machine Properties wizard.

You can reconfigure virtual machine hardware while the virtual machine is running, if the following conditions are met:

- The virtual machine has a guest operating system that supports hot-plug functionality. See the *Guest Operating System Installation Guide*.
- The virtual machine is using hardware version 7.

The following examples illustrate connecting and disconnecting a virtual device.

- The `connectdevice` option connects the virtual IDE device CD/DVD Drive 2 to the specified virtual machine.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
            /vmfs/volumes/Storage2/testvm/testvm.vmx connectdevice "CD/DVD Drive 2"
```

- The `disconnectdevice` option disconnects the virtual device.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
            /vmfs/volumes/Storage2/testvm/testvm.vmx disconnectdevice "CD/DVD Drive 2"
```

Retrieving User Input

You can use the `answer` option to prompt for user input and use the input in your script.

You might use this option when you want to configure a virtual machine based on a users's input. For example:

1 Clone a virtual machine and provide the default virtual disk type.

2 When you power on the virtual machine, it prompts for the desired virtual disk type.

Forcibly Stopping Virtual Machines

In some cases, virtual machines do not respond to the normal shutdown or stop commands. In these cases, it might be necessary to forcibly shut down the virtual machines. Forcibly shutting down a virtual machine might result in guest operating system data loss and is similar to pulling the power cable on a physical machine.

You can forcibly stop virtual machines that are not responding to normal stop operation with the `esxcli vms vm kill` command.

To forcibly stop a virtual machine

1 List all running virtual machines on the system to see the World ID of the virtual machine you want to stop.

```
esxcli vms vm list
```

2 Stop the virtual machine by running the following command.

```
esxcli vms vm kill --type <kill_type> --world-id <ID>
```

The command supports three `--type` options. Try the types sequentially (soft before hard, hard before force). The following types are supported through the `--type` option:

- `soft` – Gives the VMX process a chance to shut down cleanly (like `kill` or `kill -SIGTERM`)
- `hard` – Stops the VMX process immediately (like `kill -9` or `kill -SIGKILL`)
- `force` – Stops the VMX process when other options do not work.

If all three options do not work, reboot your ESX/ESXi host to resolve the issue.

Managing Third-Party Storage Arrays with esxcli

9

The `esxcli` command is available as a service console command (no authentication) on ESX hosts and as a vCLI for both ESX and ESXi hosts. The namespaces and commands that `esxcli` makes available depend entirely on the system on which it is installed. Run `esxcli --help` for information on availability on your system. Because `esxcli` options depend completely on the environment, no manpage is available.

IMPORTANT This chapter discusses third-party storage management options. It is not a complete reference to `esxcli`. See "Managing Path Policies with esxcli" on page 55, "Masking Paths with esxcli corestorage claimrule" on page 57, and "Managing iSCSI Sessions" on page 80 for additional uses of `esxcli`. See "esxcli Command Overview" on page 136 for a complete reference.

`esxcli` is not included in the vSphere Command-Line Reference. Use the information in this manual or the command-line help for information.

This chapter explains how to use `esxcli` 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 with the vSphere Client.

This chapter uses the following acronyms.

Acronym	Meaning
PSA	Pluggable Storage Architecture.
NMP	Native Multipathing Plugin. Generic VMware multipathing module.
PSP	Path Selection Plugin. Handles path selection for a given device.
SATP	Storage Array Type Plugin. Handles path failover for a given storage array.

The chapter includes these topics:

esxcli Command Syntax

The `esxcli` vCLI command has the following syntax:

```
esxcli <conn_options> <namespace> <app> <cmd> [cmd options]
```

Option	Description
`<conn_options>`	Connection parameters for the vCLI must precede all other parameters, or you must perform authentication in other ways. For example, you can perform authentication by using vi-fastpass on vMA, or by using environment variables. See "vCLI Connection Options" on page 23. esxcli does not support the credential store.
`<namespace>`	Namespace. Examples include the following name spaces: ■ `nmp` – VMware native multipathing commands. ■ `swiscsi` – Commands in the software iSCSI name space. ■ `corestorage` – VMware core storage commands.
`<app>`	Area within the name space to which the command applies.
`<cmd>`	Command to be called.
`<cmd options>`	Command options.

In contrast to other vCLI 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.

Command-line help for the `esxcli` vCLI is available on a per-level basis. You can call help as follows:

Command	Example	Output
`esxcli`	`esxcli <Enter>`	Lists all supported name spaces on this system.
`esxcli --help` `esxcli -?`	`esxcli --help` `esxcli -?`	Displays help for supported connection options.
`esxcli <conn_options> --help` `esxcli <conn_options> -?`	`esxcli --server S1 --help` `esxcli --server S1 -?`	Displays help for supported name spaces.
`esxcli <conn_parms> <namespace> --help` `esxcli <conn_parms> <namespace> -?`	`esxcli --server S1 nmp --help` `esxcli --server S1 nmp -?`	Displays help for supported apps for this namespace.
`esxcli <conn_options> <namespace> <app> --help` `esxcli <conn_options> <namespace> <app> -?`	`esxcli --server S1 nmp device --help` `esxcli --server S1 nmp device -?`	Displays help for supported commands for this app.
`esxcli <conn_options> <namespace> <app> <command> --help` `esxcli <conn_options> <namespace> <app> <command> -?`	`esxcli --server S1 nmp device setpolicy --help` `esxcli --server S1 nmp device setpolicy -?`	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 manage devices associated with NMP and to set path policies.

The NMP supports all storage arrays listed on the VMware storage Hardware Compatibility List (HCL) and provides a path selection algorithm based on the array type. The NMP associates a set of physical paths with a storage device (LUN). A Storage Array Type Plugin (SATP) determines how path failover is handled for a specific storage array. A Path Selection Plugin (PSP) determines which physical path is used to issue an I/O request to a storage device. SATPs and PSPs are plugins within the NMP plugin.

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`.

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.

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 `esxcli nmp psp list` to display all currently available PSPs. See Table 5-1, "Supported Path Policies," on page 55. See the *ESX Configuration Guide* and the *ESXi Configuration Guide* for a discussion of path policies.

To set the path policy for the specified device to `VMW_PSP_FIXED`, run the following command:

```
esxcli <conn_options> nmp device setpolicy --device naa.xxx --psp VMW_PSP_FIXED
```

Listing Paths with esxcli nmp path

Use the `path` option to list paths claimed by NMP. By default, the command displays information about all paths on all devices. You can filter in the following ways:

- Only show paths to a singe device (`esxcli nmp path list --device <device>`).
- Only show information for a single path (`esxcli nmp path list --path <path>` and `esxcli nmp path --device <device>`).

To list devices, call `esxcli nmp device list`.

Managing Path Selection Policy Plugins with esxcli nmp psp

Use `esxcli nmp psp` to manage VMware path selection policy plugins included with the VMware NMP plugin and to manage third-party PSPs.

IMPORTANT When used with third-party PSPs, the syntax depends on the third-party PSP implementation.

Retrieving PSP Information

The `esxcli nmp psp getconfig` command retrieves PSP configuration parameters. The type of PSP determines whether you specify `--device`, `--path`, or both.

- Use `--device` for PSPs that are set to `VMW_PSP_RR`, `VMW_PSP_FIXED` or `VMW_PSP_MRU`.
- Use `--path` for PSPs that are set to `VMW_PSP_FIXED` or `VMW_PSP_MRU`. No path configuration information is available for `VMW_PSP_RR`.

To retrieve PSP configuration parameters, run `esxcli nmp psp getconfig` with the `--device` or the `--path` option.

- Retrieve the PSP configuration for the specified device. See Table 5-1, "Supported Path Policies," on page 55.

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

- Retrieve the PSP configuration for the specified path.

  ```
  esxcli <conn_options> nmp psp getconfig --path vmhba4:C1:T2:L23
  ```

The `esxcli nmp psp list` command shows the list of Path Selection Plugins on the system and a brief description of each plugin.

Setting Configuration Parameters for Third-Party Extensions

The `esxcli nmp psp setconfig` command supports future third-party PSA expansion. The `setconfig` command sets PSP configuration parameters for those third-party extensions.

NOTE Use `esxcli nmp roundrobin setconfig` for other path policy configuration. See "Customizing Round Robin Setup with esxcli nmp roundrobin" on page 101.

The options depend on the currently set path policy.

Options	Description
`--config <config_string>` `-c <config_string>`	Configuration string to set for the device specified by `--path`. See Table 5-1, "Supported Path Policies," on page 55.
`--device <device>` `-d <device>`	Device for which you want to customize the path policy.
`--path <path>` `-p <path>`	Path for which you want to customize the path policy.

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.

Options	Description
`--device <device>` `-d <device>`	Device for which you want to get the preferred path. This device must be controlled by the `VMW_PSP_FIXED` PSP.

To return the path configured as the preferred path for the specified device, run the following command:

```
esxcli <conn_options> nmp device getpreferred --device naa.xxx
```

esxcli nmp fixed setpreferred

The `setpreferred` command sets the preferred path on a specified device that is using NMP and the `VMW_PSP_FIXED` path policy.

Options	Description
`--device <device>` `-d <device>`	Device for which you want to set the preferred path. This device must be controlled by the `VMW_PSP_FIXED` PSP. Use `esxcli nmp device --list` to list the policies for all devices.
`--path <path>` `-p <path>`	Path to set as the preferred path for the specified device.

To set the preferred path for the specified device to `vmhba3:C0:T5:L3`, run the following command:

```
esxcli <conn_options> nmp fixed setpreferred --device naa.xxx --path vmhba3:C0:T5:L3
```

Customizing Round Robin Setup with esxcli nmp roundrobin

The `roundrobin` option sets round robin path options on a device controlled by the `VMW_PSP_RR` PSP.

To specify and customize round robin path policies

1 Set the path policy to round robin.

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

2 Specify when to switch paths.

You can choose the number of I/O operations, number of bytes, and whether active unoptimized paths are used (see "esxcli nmp roundrobin setconfig" on page 101). For example:

```
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.

esxcli nmp roundrobin getconfig

The `getconfig` command retrieves path selection settings for a device that is using the `roundrobin` PSP.

Options	Description
`-d <device>` `--device <device>`	Device to get roundrobin properties for.

esxcli nmp roundrobin setconfig

The `setconfig` command specifies under which conditions a device that is using the `VMW_PSP_RR` PSP changes to a different path. You can use `--bytes` or `--iops` to specify when the path should change.

Options	Description
`--bytes` `-B`	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 `--type` is set to `bytes`.
`--device` `-d`	Device to set round robin properties for. This device must be controlled by the round robin (`VMW_PSP_RR`) PSP.

Options	Description
--iops -I	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 --type is set to iops.
--type -t	Type of round robin path switching to enable for this device. Valid values for type are: ■ bytes: Set the trigger for path switching based on the number of bytes sent down a path. ■ default: Set the trigger for path switching back to default values. ■ iops: Set the trigger for path switching based on the number of I/O operations on a path. An equal sign (=) before the type or double quotes around the type are optional.
--useANO -U	If set to 1, the round robin PSP includes paths in the active, unoptimized state in the round robin set. If set to 0, the PSP uses active, unoptimized paths only if no active optimized paths are available. Otherwise, the PSP includes only active optimized paths in the round robin path set.

Managing SATPs with esxcli nmp satp

The satp option manages SATPs and allows you to perform the following 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 FC array with a vendor and model not listed in the SATP rules is VMW_SATP_DEFAULT_AA.

Retrieving Information About SATPs

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 and displays information about these SATPs.

```
esxcli <conn_options> nmp --satp|-s list
```

The listrules command lists the claim rules for SATPs.

```
esxcli <conn_options> nmp --satp|-s listrules
```

Adding SATP Rules

Claim rules specify that a storage device that uses a certain driver or transport or has a certain vendor or model should use a certain SATP. The addrule command adds a rule that performs such a mapping to the list of claim rules. The deleterule command deletes an existing rule. The options you specify define the rule. For example, the following command specifies that if a path has vendor VMWARE and model Virtual, the PSA assigns it to the VMW_SATP_LOCAL SATP.

```
esxcli <conn_options> nmp satp addrule --satp="VMW_SATP_LOCAL" --vendor="VMWARE"
               --model="Virtual" --description="VMware virtual disk"
```

Option	Description
--driver -D	Driver string to set when adding the SATP claim rule.
--device -d	Device to set when adding SATP claim rules. Device rules are mutually exclusive with vendor/model and driver rules.
--force -f	Force claim rules to ignore validity checks and install the rule even if checks fail.
--model -M	Model string to set when adding the SATP claim rule. Can be the model name or a pattern ^mod*, which matches all devices that start with mod. That is, the pattern successfully matches mod1 and modz, but not mymod1. The command supports the start/end (^) and wildcard (*) functionality but no other regular expressions.

Option	Description
--transport -R	Transport string to set when adding the SATP claim rule. Describes the type of storage HBA, for example, iscsi or fc.
--vendor -V	Vendor string to set when adding the SATP claim rule.
--satp -s	SATP for which the rule is added.
--claim-option -c	Claim option string to set when adding the SATP claim rule.
--description -e	Description string to set when adding the SATP claim rule.
--option -o	Option string to set when adding the SATP claim rule.
--psp -P	Default PSP for the SATP claim rule.
--psp-option -O	PSP options for the SATP claim rule.

The following examples illustrate adding SATP rules.

- Add a SATP rule that specifies that disks with vendor string VMWARE and model string Virtual should be added to VMW_SATP_LOCAL.

```
esxcli <conn_options> nmp satp addrule --satp="VMW_SATP_LOCAL" --vendor="VMWARE"
    --model="Virtual" --description="VMware virtual disk"
```

- Add a SATP rule that specifies that disks with the driver string somedriver should be added to VMW_SATP_LOCAL.

```
esxcli <conn_options> nmp satp addrule --satp="VMW_SATP_LOCAL" --driver="somedriver"
```

- Add a rule that specifies that all storage devices with vendor string ABC and a model name that starts with 120 should use VMW_SATP_DEFAULT_AA (VMW_SATP_DEFAULT_AA is an example).

```
esxcli <conn_options> nmp satp addrule --satp VMW_SATP_DEFAULT_AA --vendor="ABC"
                --model="^120*
```

Deleting SATP Rules

The deleterule command deletes an existing SATP rule. The options you specify define the rule to delete. The options listed for "Adding SATP Rules" on page 102 are supported.

The following example deletes the rule that assigns devices with vendor string VMWARE and model string Virtual to VMW_SATP_LOCAL.

```
C:\WINDOWS\system32>esxcli <conn_options> nmp satp deleterule
        --satp="VMW_SATP_LOCAL" --vendor="VMWARE" --model="Virtual"
```

Retrieving and Setting SATP Configuration Parameters

The esxcli nmp satp getconfig command retrieves per-path or per-device SATP configuration parameters. For each SATP, specify either --device or --path, but not both.

IMPORTANT Not all SATPs support getconfig for devices or getconfig for paths.

Use this command to retrieve per device or per path SATP configuration parameters, and to see whether you can set certain configuration parameters for a device or path. For example:

```
# esxcli --config /my-config-file nmp satp getconfig --device naa.60019b9000dd21c500002d974a0acfa0
SATP VMW_SATP_LSI does not support device configuration.
# esxcli --config /my-config-file nmp satp getconfig -path vmhba1:C0:T0:L8
INIT,AVT OFF,v5.4,DUAL ACTIVE,ESX FAILOVER
```

The `esxcli nmp satp setconfig` command sets configuration parameters for third-party SATPs that are loaded into the system, if they support device configuration. You can set per-path or per-device SATP configuration parameters. The command sets the configuration for a specified device or path, regardless of the SATP currently associated with that device or path. VMware supports the following configuration strings. Other strings might be supported by a third-party SATP.

Options	Description
`--config` `-c`	Configuration string to set for the path specified by `--path` or the device specified by `--device`. You can set the configuration for the following SATPs: ■ VMW_SATP_ALUA_CX ■ VMW_SATP_ALUA ■ VMW_SATP_CX ■ VMW_SATP_INV You can specify one of the following device configuration strings: ■ `navireg_on` – starts automatic registration of the device with Navisphere. ■ `navireg_off` – stops the automatic registration of the device. ■ `ipfilter_on` – stops the sending of the host name for Navisphere registration. Used if host is known as `localhost`. ■ `ipfilter_off` – enables the sending of the host name during Navisphere registration.
`--device` `-d`	Device to set SATP configuration for. Not all SATPs support the `setconfig` option on devices.
`--path` `-p`	Path to set SATP configuration for. Not all SATPs support the `setconfig` option on paths.

Setting the Default PSP

The `esxcli nmp satp setdefault` command sets the default PSP for a specified SATP.

Options	Description
`--psp` `-P`	Default path selection policy to set for the SATP specified by `--satp`.
`--satp` `-s`	SATP name for the plugin for which you want to set the default PSP.

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 and reclaim actions fail.

IMPORTANT The help 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 a device is in use. Only PSA plugin developers or administrators who troubleshoot PSA plugins use this command.

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, disassociating 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 outstanding requests. You cannot unclaim the ESXi USB partition or devices with VMFS volumes on them. It is therefore normal for this command to fail, especially when you specify a plugin or adapter to unclaim.

Unclaiming does not persist. Periodic path claiming reclaims unclaimed paths unless claim rules are configured to mask a path (see "Masking Paths with esxcli corestorage claimrule" on page 57).

IMPORTANT The `unclaim` command unclaims paths associated with a device. You cannot use this command to unclaim paths associated with the MASK_PATH plugin because those paths are not associated with a device.

Options	Description
`--adapter <adapter>` `-A <adapter>`	If `--type` is set to `location`, specifies the name of the HBA for the paths that you want to unclaim. If you do not specify this option, unclaiming runs on paths from all adapters.
`--channel <channel>` `-C <channel>`	If `--type` is set to `location`, specifies the SCSI channel number for the paths that you want to unclaim. If you do not specify this option, unclaiming runs on paths from all channels.
`--claimrule-class <cl>` `-c <cl>`	Claim rule class to use in this operation. You can specify MP (Multipathing), Filter, or VAAI. Multipathing is the default. Filter is used only for VAAI. Specify claim rules for both VAAI_FILTER and VAAI plugin to use it.
`--device <device>` `-d <device>`	If `--type` is set to `device`, 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 `driver`, unclaims all paths specified by this HBA driver.
`--lun <lun_number>` `-L <lun_number>`	If `--type` is `location`, specifies the SCSI LUN for the paths to unclaim. If you do not specify `--lun`, unclaiming runs on paths with any LUN number.
`--model <model>` `-m <model>`	If `--type` is `vendor`, attempts to unclaim all paths to devices with specific model information (for multipathing plugins) or unclaim the device itself (for filter plugins). If there are active I/O operations on this device, at least one path fails to unclaim.
`--path <path>` `-p <path>`	If `--type` is `path`, unclaims a path specified by its path UID or runtime name.
`--plugin <plugin>` `-P`	If `--type` is `plugin`, 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 `location`, 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 `location`, `path`, `driver`, `device`, `plugin`, and `vendor`.
`--vendor <vendor>` `-v <vendor>`	If `--type` is `vendor`, attempts to unclaim all paths to devices with specific vendor info (for multipathing plugins) or unclaim the device itself (for filter plugins). If there are any active I/O operations on this device, at least one path fails to unclaim

The following troubleshooting command unclaims all paths to `vmhba1`. Run `vicfg-mpath <conn_options> -l` to verify the command succeeded.

```
esxcli <conn_options> corestorage claiming unclaim --type location -A vmhba1
```

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.

To change the current claim rules in the VMkernel

1 Run one of the `esxcli corestorage claimrule` modification commands (`add`, `remove`, or `move`).

2 Run `esxcli corestorage claimrule load` to replace the current rules in the VMkernel with the modified rules from the configuration file.

You can also run `esxcli corestorage claimrule pluginlist` to list all plugins.

Adding Claim Rules with 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 "Masking Paths with esxcli corestorage claimrule" on page 57. You must load the rules after you add them.

Options	Description
`--adapter <adapter>` `-A <adapter>`	Adapter of the paths to use. Valid only if `--type` is `location`.
`--autoassign` `-u`	Adds a claim rule based on its characteristics. The rule number is not required.
`--claimrule-class <cl>` `-c <cl>`	Claim rule class to use in this operation. You can specify MP (default), Filter, or VAAI. To configure hardware acceleration for a new array, add two claim rules, one for the VAAI filter and another for the VAAI plugin. See the *Fibre Channel SAN Configuration Guide* and the *iSCSI SAN Configuration Guide* for detailed instructions.
`--channel <channel>` `-C <channel>`	Channel of the paths to use. Valid only if `--type` is `location`.
`--driver <driver>` `-D <driver>`	Driver for the HBA of the paths to use. Valid only if `--type` is `vendor`.
`--force` `-f`	Force claim rules to ignore validity checks and install the rule.
`--lun <lun_number>` `-L <lun_number>`	LUN of the paths to use. Valid only if `--type` is `location`.
`--model <model>` `-M <model>`	Model of the paths to use. Valid only if `--type` is `vendor`. Valid values are values of the `Model` string from the SCSI inquiry string. Run `vicfg-scsidevs <conn_options> -l` on each device to see model string values.
`--plugin` `-P`	PSA plugin to use. Currently, the values are `NMP` or `MASK_PATH`, but third parties can ship their own PSA plugins in the future. `MASK_PATH` refers to the plugin `MASK_PATH_PLUGIN`. The command adds claimrules for this plugin if the user wants to mask the path. ESX 3.5 includes the `MaskLUNs` advanced configuration option. This option is not available in ESX/ESXi 4. It has been replaced by the `MASK_PATH_PLUGIN`. You can add a claim rule that causes the `MASK_PATH_PLUGIN` to claim the path to mask a path or LUN from the ESX/ESXi host. See "Masking Paths with esxcli corestorage claimrule" on page 57.
`--rule <rule_ID>` `-r <rule_ID>`	Rule ID to use. Run `esxcli corestorage claimrule list` 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.
`--target <target>` `-T <target>`	Target of the paths to use. Valid only if `--type` is `location`.

Options	Description
`--transport <transport>` `-R <transport>`	Transport of the paths to use. Valid only if `--type` is `transport`. The following values are supported: ■ `block` – block storage connection ■ `fc` – FibreChannel transmission ■ `iscsivendor` – iSCSI connection ■ `iscsi` – not currently used ■ `ide` – IDE storage connection ■ `sas` – SAS storage connection ■ `sata` – SATA storage connection ■ `usb` – USB storage connection ■ `parallel` – parallel transmission ■ `unknown`
`--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 `vicfg-scsidevs <conn_options> -l` on each device to see vendor string values.

The following examples illustrate adding claim rules:

- Add 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 321 -t location -A vmhba0 -C 0 -T 0 -L 0
             -P NMP
```

- Add rule 429, which claims all paths provided by an adapter with the mptscsi driver for the `MASK_PATH` plugin.

```
esxcli <conn_options> corestorage claimrule add -r 429 -t driver -D mptscsi -P MASK_PATH
```

- Add rule 914, which claims all paths with vendor string `VMWARE` and model string `Virtual` for the NMP plugin.

```
esxcli <conn_options> corestorage claimrule add -r 914 -t vendor -V VMWARE -M Virtual -P NMP
```

- Add rule 1015, which claims all paths provided by FC adapters for the NMP plugin.

```
esxcli <conn_options> corestorage claimrule add -r 1015 -t transport -R fc -P NMP
```

Converting ESX 3.5 LUN Masks to Claim Rule Format

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.

To convert ESX 3.5 format LUN masks to claim rule format

1 Run `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 Run `esxcli corestorage claimrule convert --commit` to actually commit the change.

When you convert LUN masking to the claim rule format after an upgrade from ESX/ESXi 3.5 to ESX/ESXi 4.x, this command converts the `/adv/Disk/MaskLUNs` advanced configuration entry in the `esx.conf` file to claim rules with `MASK_PATH` as the plug-in.

IMPORTANT This conversion does not work for all input Mask LUN variations. For example, role conversion for software iSCSI LUNs is not supported.

Inspect the list of generated claim rules carefully before you commit them by using `--commit`.

Table 9-1. 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.

Deleting Claim Rules with 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.

Option	Description
`--rule <rule_ID>` `-r <rule_ID>`	ID of the rule to be deleted. Run `esxcli corestorage claimrule list` to see the rule ID.

The following example deletes rule 1015.

```
esxcli <conn_options> corestorage claimrule delete -r 1015
```

Listing Claim Rules with esxcli corestorage claimrule list

The `list` command lists all claim rules on the system. You can specify the claim rule class as an argument.

Option	Description
`--claimrule-class <cl>` `-c <cl>`	Claim rule class to use in this operation. You can specify `MP` (Multipathing), `Filter`, or `VAAI`. Multipathing is the default. Filter is used only for VAAI. Specify claim rules for both `VAAI_FILTER` and `VAAI` plugin to use it.

You can run the command as follows. The equal sign is optional, so both forms of the command have the same result.

```
esxcli --config /vmc-store09 corestorage claimrule list -c Filter
esxcli --config /vmc-store09 corestorage claimrule list --claimrule-class=Filter
```

Loading Claim Rules with 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`.

Moving Claim Rules with esxcli corestorage claimrule move

The `move` command moves a claim rule from one rule ID to another.

Options	Description
`--claimrule-class <cl>` `-c <cl>`	Claim rule class to use in this operation.
`--new-rule <rule_ID>` `-n <rule_ID>`	New rule ID you want to give to the rule specified by the `--rule` option.
`--rule <rule_ID>` `-r <rule_ID>`	ID of the rule to be deleted. Run `esxcli corestorage claimrule list` to display the rule ID.

The following example renames rule 1016 to rule 1015 and deletes rule 1016.

```
esxcli <conn_options> corestorage claimrule move -r 1015 -n 1016
```

esxcli corestorage claimrule run

The `run` command runs path claiming rules. This command is for troubleshooting and boot time configuration.

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.
`--claimrule-class` `-c`	Claim rule class to use in this operation.
`--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`.

Managing vSphere Networking 10

The vSphere CLI networking commands allow you to manage the vSphere network services. You can connect virtual machines to the physical network and to each other and configure vNetwork Standard Switches and vNetwork Distributed Switches. You can also set up your vSphere environment to work with external networks such as SNMP or NTP.

This chapter includes the following topics:

Introduction to vSphere Networking

At the core of vSphere Networking are virtual switches. vSphere supports standard switches (vSS) and vNetwork Distributed Switch (vDS). Each virtual switch has a preset number of ports and one or more port groups.

Virtual switches allow your virtual machines to connect to each other and to connect to the outside world.

- When two or more virtual machines are connected to the same virtual switch, network traffic between them is routed locally.
- When virtual machines are connected to a virtual switch that is connected to an uplink adapter, each virtual machine can access the external network through that uplink. The adapter can be an uplink connected to a vSS or a dvUplink connected to a vDS.

Virtual switches allow your ESX/ESXi host to migrate virtual machines with VMware VMotion and to use IP storage through VMkernel network interfaces.

- Using VMotion, you can migrate running virtual machines with no downtime. You can enable VMotion with `vicfg-vmknic --enable-vmotion`.
- IP storage refers to any form of storage that uses TCP/IP network communication as its foundation, which includes iSCSI and NFS for ESX/ESXi. Because these storage types are network based, they can use the same VMkernel interface and port group.

The network services that the VMkernel provides (iSCSI, NFS, and VMotion) use a TCP/IP stack in the VMkernel. This TCP/IP stack is completely separate from the TCP/IP stack used in the ESX service console. The VMkernel TCP/IP stack is also separate from the guest operating system's network stack. Each of these stacks accesses various networks by attaching to one or more port groups on one or more virtual switches.

Networking Using vNetwork Standard Switches

vNetwork Standard Switches (vSS) allow you to connect virtual machines to the outside world.

Figure 10-1. Networking with vNetwork Standard Switches

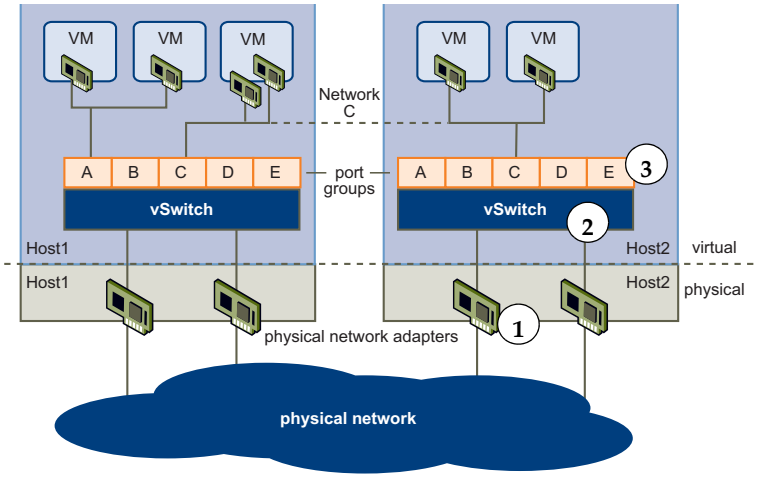


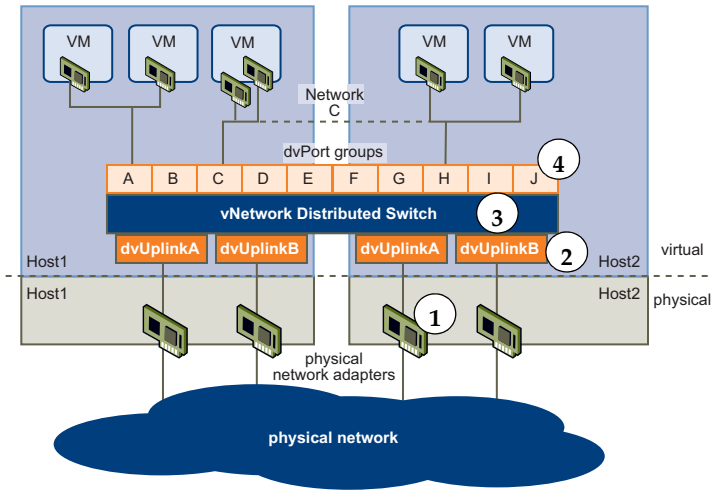
Figure 10-1 shows the relationship between the physical and virtual network elements. The numbers match those in the figure.

- Associated with each ESX/ESXi host are one or more uplink adapters (1). Uplink adapters represent the physical switches the ESX/ESXi host uses to connect to the network. You can manage uplink adapters using the `vicfg-nics` vCLI command. See "Managing Uplink Adapters with vicfg-nics" on page 116.
- Each uplink adapter is connected to a vSS (2). You can manage a vSS and associate it with uplink adapters by using the `vicfg-vswitch` vCLI command. See "Setting Up Virtual Switches and Associating a Switch with a Network Interface" on page 114.
- Associated with the vSS are port groups (3). Port group is a unique concept in the virtual environment. You can configure port groups to enforce policies that provide enhanced networking security, network segmentation, better performance, high availability, and traffic management. You can use the `vicfg-vswitch` command to associate a vSS with a port group, and the `vicfg-vmknic` command to associate a port group with a VMkernel network interface.
- The VMkernel TCP/IP networking stack supports iSCSI, NFS, and VMotion and has an associated VMkernel network interface. You configure VMkernel network interfaces with `vicfg-vmknic`. See "Adding and Modifying VMkernel Network Interfaces with vicfg-vmknic" on page 117. Separate VMkernel network interfaces are often used for separate tasks, for example, you might devote one VMkernel Network interface card to VMotion only. Virtual machines run their own systems' TCP/IP stacks and connect to the VMkernel at the Ethernet level through virtual switches.

Networking Using vNetwork Distributed Switches

When you want to connect a virtual machine to the outside world, you can use a vSS or a vDS. With a vDS, the virtual machine can maintain its network settings even if it is migrated to a different host.

Figure 10-2. Networking with vNetwork Distributed Switches



- Each physical adapter (1) on the host is paired with a dvUplink adapter (2), which represents the uplink to the virtual machine. With vDS, the virtual machine no longer depends on the host's physical uplink but on the (virtual) dvUplink. You can manage a dvUplink using the vSphere Client or `vicfg-vswitch`.
- The vDS itself (3) functions as a single virtual switch across all associated hosts. Because the switch is not associated with a single host, virtual machines can maintain consistent network configuration as they migrate from one host to another.

  Like a vSS, each vDS is a network hub that virtual machines can use. A vDS can route traffic internally between virtual machines or link to an external network by connecting to physical network adapters. You create a vDS using the vSphere Client UI, but can manage some aspects of a vDS using `vicfg-vswitch`. See "Setting Up Virtual Switches and Associating a Switch with a Network Interface" on page 114.
- Each vDS can have one or more dvPort groups (4) assigned to it. dvPort groups aggregate multiple ports under a common configuration and provide a stable anchor point for virtual machines connecting to labeled networks.
- Just as a vSS, vDS supports using one or more VMkernel network interfaces. Each VMkernel network interface can manage part of the traffic handled by the ESX/ESXi host. See "Adding and Modifying VMkernel Network Interfaces with vicfg-vmknic" on page 117.

Setting Up vSphere Networking with vNetwork Standard Switches

You can set up your virtual network by performing these tasks, discussed in more detail in this chapter:

1. Create or manipulate virtual switches using `vicfg-vswitch`. By default, each ESX/ESXi host has one virtual switch, vSwitch0. You can create additional virtual switches or manage existing switches. See "Setting Up Virtual Switches and Associating a Switch with a Network Interface" on page 114.
2. (Optional) Make changes to the uplink adapter using `vicfg-nics`. See "Managing Uplink Adapters with vicfg-nics" on page 116.
3. (Optional) Use `vicfg-vswitch` to add port groups to the virtual switch. See "Checking, Adding, and Removing Port Groups" on page 115.
4. (Optional) Use `vicfg-vswitch` to establish VLANs by associating port groups with VLAN IDs. See "Setting the Port Group VLAN ID" on page 116.
5. Use `vicfg-vmknic` to configure the VMkernel network interfaces. See "Adding and Modifying VMkernel Network Interfaces with vicfg-vmknic" on page 117.

Setting Up Virtual Switches and Associating a Switch with a Network Interface

A virtual switch models a physical Ethernet switch. You can manage virtual switches and port groups using the vSphere Client (see the *ESX Configuration Guide* and the *ESXi Configuration Guide*) or using vSphere CLI commands.

You can create a maximum of 127 virtual switches on a single ESX/ESXi host. By default, each ESX/ESXi host has a single virtual switch called `vSwitch0`. A virtual switch has 56 logical ports by default. See the *Configuration Maximums* document on the vSphere documentation main page for details. Ports connect to the virtual machines and the ESX/ESXi physical adapters.

- You can connect one virtual machine network adapter to each port using the vSphere Client UI.
- You can connect the uplink adapter to the virtual switches using `vicfg-vswitch`. See "Linking and Unlinking Uplink Adapters" on page 116.

When two or more virtual machines are connected to the same virtual switches, network traffic between them is routed locally. If an uplink adapter is attached to the virtual switches, each virtual machine can access the external network that the adapter is connected to.

This section discusses working in a standard vSwitch (vSS) environment. See "Managing vNetwork Distributed Switches" on page 119 for information about vDS environments.

When working with virtual switches and port groups, perform the following tasks:

1. Find out which virtual switches are available and (optionally) what the associated MTU and CDP (Cisco Discovery Protocol) settings are. See "Retrieving Information about Virtual Switches" on page 114.
2. Add a virtual switch. See "Adding and Deleting Virtual Switches" on page 115.
3. For a newly added switch, perform these tasks:
   a. Add a port group. See "Checking, Adding, and Removing Port Groups" on page 115.
   b. (Optional) Set the port group VLAN ID. See "Setting the Port Group VLAN ID" on page 116.
   c. Add an uplink adapter. See "Linking and Unlinking Uplink Adapters" on page 116.
   d. (Optional) Change the MTU or CDP settings. See "Setting Switch Attributes" on page 115.

Retrieving Information about Virtual Switches

You can retrieve information about virtual switches by using the following `vcifg-vswitch` options:

- Check whether `vSwitch1` exists.

  ```
  vicfg-vswitch <conn_options> -c vSwitch1
  ```

  The command returns 1 if the switch exists, 0 if the switch does not exist.

- List all virtual switches and associated port groups.

  ```
  vicfg-vswitch <conn_options> -l
  ```

  The command prints information about the virtual switch, which might include its name, number of ports, MTU, port groups, and other information. The precise information depends on the target system and on whether you list information for a vSS or a vDS.

  For ESXi systems, the default port groups are `Management Network` and `VM Network`. For ESX systems, the default port groups are `Service console` and `VM Network`.

- Retrieve the current CDP (Cisco Discovery Protocol) setting for this virtual switch.

  If CDP is enabled on a virtual switch, ESX/ESXi administrators can find out which Cisco switch port is connected to which virtual switch uplink. CDP is a link-level protocol that supports discovery of CDP-aware network hardware at either end of a direct connection. CDP is bit forwarded through switches. CDP is a simple advertisement protocol which beacons information about the switch or host and some port information.

  ```
  vicfg-vswitch <conn_options> --get-cdp vSwitch1
  ```

Adding and Deleting Virtual Switches

You can add and delete virtual switches using the `--add|-a` and `--delete|-d` options.

- Add a virtual switch.

  ```
  vicfg-vswitch <conn_options> --add vSwitch2
  ```

  After you have added a virtual switch, you can set switch attributes ("Setting Switch Attributes" on page 115) and add one or more uplink adapters ("Linking and Unlinking Uplink Adapters" on page 116).

- Delete a virtual switch.

  ```
  vicfg-vswitch <conn_options> --delete vSwitch1
  ```

  You cannot delete a virtual switch if any ports on the switch are still in use by VMkernel networks, virtual machines, or `vswifs`. Run `vicfg-vswitch --list` to determine whether a virtual switch is in use.

Setting Switch Attributes

You can set the maximum transmission unit (MTU) and CDP status for a virtual switch. The CDP status shows which Cisco switch port is connected to which uplink.

- Set the MTU for a vSwitch.

  ```
  vicfg-vswitch <conn_options> -m 9000 vSwitch1
  ```

  The MTU is the size (in bytes) of the largest protocol data unit the switch can process. When you set this option, it affects all uplinks assigned to the virtual switch.

- Set the CDP value for a vSwitch. You can set status to `down`, `listen`, `advertise`, or `both.`

  ```
  vicfg-vswitch <conn_options> --set-cdp 'listen'
  ```

Checking, Adding, and Removing Port Groups

Network services connect to vSwitches through port groups. A port group allows you to group traffic and specify configuration options such as bandwidth limitations and VLAN tagging policies for each port in the port group. A virtual switch must have one port group assigned to it. You can assign additional port groups.

You can use `vicfg-vswitch` to check, add, and remove port groups.

- Check whether port groups are currently associated with a vSwitch.

  ```
  vicfg-vswitch <conn_options> --check-pg <port_group> vSwitch1
  ```

  The command returns 0 if the specified port group is associated with the vSwitch, and returns 1 otherwise. Use `vicfg-vswitch --list` to list all port groups.

- Add a port group.

  ```
  vicfg-vswitch <conn_options> --add-pg <port_group_name> vSwitch1
  ```

- Delete one of the existing port groups.

  ```
  vicfg-vswitch <conn_options> --del-pg <port_group_name> vSwitch1
  ```

Connecting and Disconnecting Uplink Adapters and Port Groups

If your setup includes one or more port groups, you can associate each port group with one or more uplink adapters (and remove the association). This functionality allows you to filter traffic from a port group to a specific uplink, even if the virtual switch is connected with multiple uplinks.

- Connect a port group with an uplink adapter.

  ```
  vicfg-vswitch --add-pg-uplink <adapter_name> --pg <port_group> <vswitch_name>
  ```

  This command fails silently if the uplink adapter does not exist.

- Remove a port group from an uplink adapter.

  ```
  vicfg-vswitch --del-pg-uplink <adapter_name> --pg <port_group> <vswitch_name>
  ```

Setting the Port Group VLAN ID

VLANs allow you to further segment a single physical LAN segment so that groups of ports are isolated as if they were on physically different segments. The standard is IEEE 802.1Q.

A VLAN ID restricts port group traffic to a logical Ethernet segment within the physical network.

- Set the VLAN ID to 4095 to allow a port group to reach port groups located on other VLAN.
- Set the VLAN ID to 0 to disable the VLAN for this port group.

If you use VLAN IDs, you must change the port group labels and VLAN IDs together so that the labels properly represent connectivity. VLAN IDs are optional.

You can use the following commands for VLAN management:

- Allow all port groups to reach port groups located on other VLANs.

  ```
  vicfg-vswitch <conn_options> --vlan 4095 --pg "ALL" vSwitch2
  ```

- Disable VLAN for port group g42

  ```
  vicfg-vswitch <conn_options> --vlan 0 --pg g42 vSwitch2
  ```

Run `vicfg-vswitch -l` to retrieve information about VLAN IDs currently associated with the vSwitches in the network.

Linking and Unlinking Uplink Adapters

When you create vSwitch using `vicfg-vswitch --add`, all traffic on that vSwitch is initially confined to that vSwitch. All virtual machines connected to the vSwitch can talk to each other, but the virtual machines cannot connect to the network or to virtual machines on other hosts. A virtual machine also cannot connect to virtual machines connected to a different vSwitch on the same host.

Having a vSwitch that is not connected to the network might make sense if you want a group of virtual machines to be able to communicate with each other, but not with other hosts or with virtual machines on other hosts. In most cases, you set up the vSwitch to transfer data to external networks by attaching one or more uplink adapters to the vSwitch.

You can use the following commands to add and remove uplink adapters:

- Add a new uplink adapter to a virtual switch.

  ```
  vicfg-vswitch <conn_options> --link vmnic15 vSwitch0
  ```

- Remove an uplink adapter from a virtual switch.

  ```
  vicfg-vswitch <conn_options> --unlink vmnic15 vSwitch0
  ```

Managing Uplink Adapters with vicfg-nics

The `vicfg-nics` command manages uplink adapters, which represent the physical NICs that connect the ESX/ESXi host to the network.

You can use `vicfg-nics` to list information and to specify speed and duplex setting for the uplink.

The following example scenario lists an uplink adapter's properties, changes the duplex and speed, and sets the uplink to auto-negotiate its speed and duplex settings.

To manipulate uplink adapter setup

1 List settings.

  ```
  vicfg-nics <conn_options> -l
  ```

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

2 Set the settings for `vmnic0` to `full` and the `speed` to 100.

  ```
  vicfg-nics <conn_options> -d full -s 100 vmnic0
  ```

3 Set `vmnic2` to auto-negotiate its speed and duplex settings.

```
vicfg-nics <conn_options> -a vmnic2
```

Adding and Modifying VMkernel Network Interfaces with vicfg-vmknic

VMkernel network interfaces are used primarily for management traffic, which can include vMotion, IP Storage, and other management traffic on the ESX/ESXi system. You can also bind a newly created VMkernel network interface for use by software and dependent hardware iSCSI using the `esxcli swiscsi nic add` command. See "esxcli swiscsi Namespace" on page 141.

The VMkernel network interface is separate from the virtual machine network. The guest operating system and application programs communicate with a VMkernel network interface through a commonly available device driver or a VMware device driver optimized for the virtual environment. In either case, communication in the guest operating system occurs as it would with a physical device. Virtual machines can also communicate with a VMkernel network interface if both use the same vSwitch.

Each VMkernel network interface has its own MAC address and one or more IP addresses, and responds to the standard Ethernet protocol as would a physical NIC. The VMkernel network interface is created with TCP Segmentation Offload (TSO) enabled.

You can configure the VMkernel network interface for IPv4 (see "To add and configure a VMkernel Network Interface with IPv4" on page 117) or for IPv6 (see "To add and configure a VMkernel Network Interface with IPv6" on page 118).

To add and configure a VMkernel Network Interface with IPv4

1 Run `vicfg-vmknic --add` to add a VMkernel network interface. You must specify the IP address using `--ip`, the netmask, and the name. For the following examples, assume that VMSF-VMK-363 is a port group to which you want to add a VMkernel network interface.

```
vicfg-vmknic <conn_options> --add --ip <ip_address> -n 255.255.255.0 VMSF-VMK-363
```

You can specify the MTU setting when adding a VMkernel network interface. You cannot change that setting at a later time.

When the command completes successfully, the newly added VMkernel network interface is enabled.

2 Change the IP address as needed.

```
vicfg-vmknic <conn_options> --ip <address> VMSF-VMK-363
```

For IPv4, choose one of the following formats:

- <X.X.X.X>– Static IPv4 address.
- DHCP – Use IPv4 DHCP.

The VMkernel supports DHCP only for ESX/ESXi 4.0 and later.

3 (Optional) Enable VMotion. By default, VMotion is disabled.

```
vicfg-vmknic <conn_options> --enable-vmotion VMSF-VMK-363
```

You can later use `--disable-vmotion` to disable VMotion for this VMkernel network interface.

4 List information about all VMkernel network interfaces on the system.

```
vicfg-vmknic <conn_options> --list
```

The command displays the network information, port group, MTU, and current state for each virtual network adapter in the system.

To add and configure a VMkernel Network Interface with IPv6

1 Run `vicfg-vmknic --add` to add a VMkernel network interface. You must specify the IP address using `--ip`, the netmask, and the port group name. For the following examples, assume that VMSF-VMK-363 is a port group to which you want to add a VMkernel network interface.

You can specify the MTU setting when adding a VMkernel network interface. You cannot change that setting at a later time.

When the command completes successfully, the newly added VMkernel network interface is enabled.

2 Enable IPv6.

```
vicfg-vmknic <conn_options> --enable-ipv6 true VMSF-VMK-363
```

3 Supply an IPv6 address.

```
vicfg-vmknic <conn_options> --ip <ip_address> VMSF-VMK-363
```

For IPv6, the IP address can have one of the following formats:

- <X:X:X::/X> – Static IPv6 address
- DHCPV6 – Use DHCP IPv6 address. The VMkernel supports DHCP only for ESX/ESXi 4.0 and later.
- AUTOCONF – Use the IPv6 address advertised by the router. If you create a VMkernel network interface with AUTOCONF, an address is assigned immediately. If you add AUTOCONF to an existing vmknic, the address is added when the router sends the next router advert.

4 (Optional) Enable VMotion. By default, VMotion is disabled.

```
vicfg-vmknic <conn_options> --enable-vmotion VMSF-VMK-363
```

You can later use `--disable-vmotion` to disable VMotion again.

5 List information about all VMkernel network interfaces on the system.

```
vicfg-vmknic <conn_options> --list
```

The list contains the network information, port group, MTU, and current state for each virtual network adapter in the system.

6 You can later remove the IPv6 address and disable IPv6.

```
vicfg-vmknic <conn_options> --unset-ip <X:X:X::/X> VMSF-VMK-363
vicfg-vmknic <conn_options> --enable-ipv6 false VMSF-VMK-363
```

Setting Up vSphere Networking with vNetwork Distributed Switch

A vDS functions as a single virtual switch across all associated hosts. A vDS allows virtual machines to maintain a consistent network configuration as they migrate across multiple hosts. See "Networking Using vNetwork Distributed Switches" on page 113.

Like a vNetwork Standard Switch, each vDS is a network hub that virtual machines can use. A vDS can forward traffic internally between virtual machines or link to an external network by connecting to uplink adapters.

Each vDS can have one or more dvPort groups assigned to it. dvPort groups group multiple ports under a common configuration and provide a stable anchor point for virtual machines connecting to labeled networks. Each dvPort group is identified by a network label, which is unique to the current datacenter. A VLAN ID, which restricts port group traffic to a logical Ethernet segment within the physical network, is optional.

You can create a vDS using the vSphere Client. After you have created a vDS, you can add hosts using the vSphere Client. You can use the vSphere Client to create dvPort groups and edit vDS properties and policies. After the vDS has been set up, you can use the vSphere Client or the `vicfg-vswitch` command to add or remove dvUplink adapters.

Managing vNetwork Distributed Switches

Certain tasks, such as creating a vNetwork Distributes Switch, creating dvPort groups, and editing vDS properties and policies, can only be performed using the vSphere Client or the vSphere Web Services SDK. See the *ESX Configuration Guide*, the *ESXi Configuration Guide*, and the white paper available through the Resources link at www.vmware.com/go/networking for information about vDS and how to configure vDS using the vSphere Client.

After the vDS has been set up, you can use `vicfg-vswitch` to add or remove dvUplink adapters.

- Add a dvUplink.

  ```
  vicfg-vswitch --add-dvp-uplink <adapter_name> --dvp <DVPort_id> <dvswitch_name>
  ```

- Remove a dvUplink.

  ```
  vicfg-vswitch --del-dvp-uplink <adapter> --dvp <DVPort_id> <dvswitch_name>
  ```

Managing Standard Networking Services in the vSphere Environment

You can use vCLI commands to set up DNS, NTP, SNMP, and the default gateway for your vSphere environment.

Setting the DNS Configuration

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.

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

In network environments where a DHCP server and a DNS server are available, ESX/ESXi hosts are automatically assigned DNS names.

In network environments where automatic DNS is not available or not desirable, you can configure static DNS information, including a host name, primary name server, secondary name server, and DNS suffixes.

To set up DNS

1 Run `vicfg-dns` without command-specific options to display DNS properties for the specified server.

  ```
  vicfg-dns <conn_options>
  ```

  If DNS is not set up for the target server, the command returns an error.

2 To change the settings, use `vicfg-dns` with `--dns`, `--domain`, or `--hostname`.

  - Specify the DNS server using the `--dns` option and a comma-separated list of hosts, in order of preference.

    ```
    vicfg-dns <conn_options --dns <dns1,dns2>
    ```

  - Configure the DNS host name for the server specified by `--server` (or `--vihost`).

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

  - Configure the DNS domain name for the server specified by `--server` (or `--vihost`).

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

3 To turn on DHCP, use the `--DHCP` option.

  ```
  vicfg-dns <conn_options> --dhcp yes
  ```

To modify DNS setup for a preconfigured server

1 Run `vicfg-dns` without command-specific options to display DNS properties for the specified server.

```
vicfg-dns <conn_options>
```

The command 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.

2 If the DNS properties are set, and you want to change the DHCP settings, you must specify the virtual network adapter to use when overriding the system DNS.

- For ESX hosts, `v_nic` must be one of the service console network adapters.
- For ESXi hosts, `v_nic` must be one of the VMkernel network adapter.

Override the existing DHCP setting as follows:

```
vicfg-dns <conn_options> --dhcp yes --v_nic <vnic>
```

Adding and Starting an NTP Server

Some protocols, such as Kerberos, must have accurate information about the current time. In those cases, you can add an NTP (Network Time Protocol) server to your ESX/ESXi host.

To manage an NTP Server

1 Run `vicfg-ntp --add` to add an NTP server to the host specified in <conn_options>. You use a host name or IP address to specify an already running NTP server.

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

2 Run `vicfg-ntp --start` to start the service.

```
vicfg-ntp <conn_options> --start
```

3 Run to `vicfg-ntp --list` to list the service.

```
vicfg-ntp <conn_options> --list
```

4 Run `vicfg-ntp --stop` to stop the service.

```
vicfg-ntp <conn_options> --stop
```

5 Run `vicfg-ntp --delete` to remove the specified NTP server from the host specified in <conn_options>.

```
vicfg-ntp <conn_options> --delete 192.XXX.XXX.XX
```

Managing the IP Gateway

If you move your ESX/ESXi host to a new physical location, you might have to change the default IP gateway. You can use the `vicfg-route` command to manage the default gateway for the VMkernel IP stack. `vicfg-route` supports a subset of the Linux `route` command's options.

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

To add, view, and delete a route entry

1 Add a route entry to the VMkernel and make it the default.

- For IPv4 networks, no additional options are required.

```
vicfg-route <conn_options> --add <network_ip> <netmask_IP> <gateway_ip>
```

For example, to add a route to 192.168.100.0 through 192.168.0.1:

```
vicfg-route -a 192.168.100.0/24 192.168.0.1
```

or

```
vicfg-route -a 192.168.100.0 255.255.255.0 192.168.0.1
```

- For IPv6 networks, use `--family v6`

  ```
  vicfg-route -f V6 --add <network_ip_and_mask> <gateway_ip>
  ```

  For example:

  ```
  vicfg-route -f V6 --add 2001:10:20:253::/64 2001:10:20:253::1
  ```

2 List route entries to check that your route was added by running the command without options.

```
vicfg-route <conn_options>
```

The output lists all networks and corresponding netmasks and gateways.

3 Set the default gateway.

- For IPv4, use this syntax:

  ```
  vicfg-route 192.168.0.1
  or
  vicfg-route -a default 192.168.0.1
  ```

- For IPv6, use this syntax:

  ```
  vicfg-route -f V6 -a default 2001:10:20:253::1
  ```

4 Run `vicfg-route --delete` to delete the route. Specify first the gateway, and then the network.

```
vicfg-route <conn_options> -d 192.168.100.0/24 192.168.0.1
```

Using vicfg-ipsec for Secure Networking

You can use `vicfg-ipsec` to set up IPsec (Internet Protocol Security), which secures IP communications coming from and arriving at ESX/ESXi hosts. Administrators who perform IPsec setup must have a solid understanding of both IPv6 and IPsec.

ESX/ESXi hosts support IPsec only for traffic using IPv6. IPv4 is not supported.

IMPORTANT In ESX/ESXi 4.1, IPv6 is by default disabled. You can turn on IPv6 by running this vCLI command:

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

You cannot run `vicfg-ipsec` with a vCenter Server system as the target (using the `--vihost` option).

The VMware implementation of IPsec adheres to the following IPv6 RFCs:

- 4301 Security Architecture for the Internet Protocol
- 4303 IP Encapsulating Security Payload (ESP)
- 4835 Cryptographic Algorithm Implementation Requirements for ESP
- 2410 The NULL Encryption Algorithm and Its Use With IPsec
- 2451 The ESP CBC-Mode Cipher Algorithms
- 3602 The AES-CBC Cipher Algorithm and Its Use with IPsec
- 2404 The Use of HMAC-SHA-1-96 within ESP and AH
- 4868 Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512

Using IPsec with ESX/ESXi

When you set up IPsec on an ESX/ESXi host, you enable protection of incoming or outgoing data. What happens precisely depends on how you set up the system's Security Associations (SAs) and Security Policies (SPs).

- An SA determines how the system protects traffic. When you create an SA, you specify the source and destination, authentication, and encryption parameters, and an identifier for the SA in the following options to `vicfg-ipsec`.
  - `sa-src` and `sa-dst`
  - `spi` (security parameter index)
  - `sa-mode` (tunnel or transport)
  - `ealgo` and `ekey`
  - `ialgo` and `ikey`
- An SP identifies and selects traffic that must be protected. An SP consists of two logical sections, a selector, and an action.

  The selector is specified by the following options to `vicfg-ipsec`.
  - `src-addr` and `src-port`
  - `dst-addr` and `dst-port`
  - `ulproto`
  - `direction` (`in` or `out`)

  The action is specified by the following options to `vicfg-ipsec`.
  - `sa-name`
  - `sp-name`
  - `action` (`none`, `discard`, `ipsec`)

Because IPsec allows you to target precisely which traffic should be encrypted, it is well suited for securing your vSphere environment. For example, you can set up the environment so all VMotion traffic is encrypted.

Managing Security Associations with vicfg-ipsec

You can specify an SA and request that the VMkernel use that SA. The following options for SA setup are supported.

Option	Description
`sa-src <source_IP>`	Source IP for the SA.
`sa-dst <destination_IP>`	Destination IP for the SA.
`spi`	Security Parameter Index (SPI) for the SA. Must be a hexadecimal number with a `0x` prefix. When IPsec is in use, ESX/ESXi uses the ESP protocol (RFC 43030), which includes authentication and encryption information and the SPI. The SPI identifies the SA to use at the receiving host. Each SA you create must have a unique combination of source, destination, protocol, and SPI.
`sa-mode [tunnel | transport]`	Either tunnel or transport. In tunnel mode, the original packet is encapsulated in another IPv6 packet, where source and destination addresses are the SA endpoint addresses.
`ealgo [null | 3des-cbc | aes128-cbc]`	Encryption algorithm to use. Choose `3des-cbc` or `aes128-cbc`, or `null` for no encryption.

Option	Description
`ekey <key>`	Encryption key to be used by the encryption algorithm. A series of hexadecimal digits with a 0x prefix or an ASCII string.
`ialgo [hmac-sha1 | hmac-sha2-256 ]`	Authentication algorithm to be used. Choose `hmac-sha1` or `hmac-sha2-256`.
`ikey`	Authentication key to be used. A series of hexadecimal digits or an ASCII string.

You can perform these main tasks with SAs:

- Create an SA with `vicfg-ipsec --add-sa`. You specify the source and destination, and the authentication mode. You also specify the authentication algorithm and authentication key to use. You must specify an encryption algorithm and key, but you can specify `null` if you want no encryption. Authentication is required and cannot be `null`. The following example includes extra line breaks for readability. The last option (`sa_2` in the example) is the name of the SA.

```
vicfg-ipsec --add-sa
            --sa-src 2001:DB8:1::121
            --sa-dst 2001:DB8:1::122
            --sa-mode transport
            --spi 0x1000
            --ealgo 3des-cbc
            --ekey 0x697076367265616479 6c6f676f336465736362636f757432
            --ialgo hmac-sha1
            --ikey 0x697076367265616479 6c6f67736861316f757432
            sa_2
```

- List an SA with `vicfg-ipsec --list-sa`. This command returns SAs currently available for use by an SP. The list includes SAs you created using `vicfg-ipsec`.
- Remove a single SA with `vicfg-ipsec --remove-sa`. If the SA is in use when you run this command, the command cannot perform the removal.
- Remove all SAs with `vicfg-ipsec --flush-sa`. This option removes all SAs even when they are in use.

CAUTION Running `vicfg-ipsec --flush-sa` removes all SAs on your system and might leave your system in an inconsistent state.

Managing Security Policies with vicfg-ipsec

After you have created one or more SAs, you can add security policies (SPs) to your ESX/ESXi hosts. While the SA specifies the authentication and encryption parameters to use, the SP identifies and selects traffic.

The following options for SP management are supported.

Option	Description
`sp-src <ip>/<p_len>`	Source IP address and prefix length.
`sp-dst <ip>/<p_len>`	Destination IP address and prefix length.
`src-port <port>`	Source port (0-65535). Specify `any` for any ports.
`dst-port <port>`	Destination port (0-65535). Specify `any` for any ports. If `ulproto` is `icmp6`, this number refers to the icmp6 type. Otherwise, this number refers to the port.
`ulproto [any | tcp | udp | icmp6]`	Upper layer protocol. Use this option to restrict the SP to only certain protocols, or use `any` to apply the SP to all protocols.
`dir [in | out]`	Direction in which you want to monitor the traffic. To monitor traffic in both directions, create two policies.

Option	Description
`action [none | discard | ipsec]`	Action to take when traffic with the specified parameters is encountered. `none` -- Take no action, that is, allow traffic unmodified. `discard` -- Do not allow data in or out. `ipsec` -- Use the authentication and encryption information specified in the SA to determine whether the data come from a trusted source.
`sp-mode [tunnel | transport]`	Mode, either `tunnel` or `transport`.
`sa-name`	Name of the SA to use by this SP.

You can perform these main tasks with SPs:

- Create an SP with `vicfg-ipsec --add-sp`. You identify the data to monitor by specifying the selector's source and destination IP address and prefix, source port and destination port, upper layer protocol, direction of traffic, action to take, and SP mode. The last two option are the name of the SA to use and the name of the SP that is being created. The following example includes extra line breaks for readability.

```
vicfg-ipsec --add-sp
    --sp-src=2001:0DB8:0001:/48
    --sp-dst=2001:0DB8:0002:/48
    --src-port=23
    --dst-port=25
    --ulproto=tcp
    --dir=out
    --action=ipsec
    --sp-mode=transport
    --sp-name sp_2
```

- List an SP with `vicfg-ipsec --list-sp`. This command returns SPs currently available. All SPs are created by the administrator.
- Remove an SP with `vicfg-ipsec --remove-sp`. If the SP is in use when you run this command, the command cannot perform the removal. You can run `vicfg-ipsec --flush-sp` instead to remove the SP even when it is in use.

CAUTION Running `vicfg-ipsec --flush-sp` removes all SPs on your system and might leave your system in an inconsistent state.

11 Monitoring ESX/ESXi Hosts

Starting with the vSphere 4.0 release, the vCenter Server makes performance charts for CPU, memory, disk I/O, networking, and storage available. You can view these performance charts by using the vSphere Client and read about them in the *Datacenter Administration Guide*. You can also perform some monitoring of your ESX/ESXi hosts using vCLI commands.

This chapter includes these topics:

Using resxtop for Performance Monitoring

The `resxtop` vCLI 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 `resxtop` and provides information about available commands and display statistics.

IMPORTANT `resxtop` is supported only on Linux.

Managing Diagnostic Partitions with vicfg-dumppart

Your host must have a diagnostic partition (dump partition) to store core dumps for debugging and for use by VMware technical support. The VMware knowledge base article at http://kb.vmware.com/kb/1004128 explains how to collect the information.

You can use the vSphere Client to create the diagnostic partition on a local disk or on a private or shared SAN LUN. You cannot use `vicfg-dumppart` to create the diagnostic partition. The SAN LUN can be set up with FibreChannel or hardware iSCSI. SAN LUNs accessed through a software iSCSI initiator are not supported.

Each host must have a diagnostic partition of 100MB. If multiple hosts share a SAN LUN, configure a diagnostic partition with 100MB for each host.

CAUTION If two hosts that share a diagnostic partition fail and save core dumps to the same slot, the core dumps might be lost. To collect core dump data, reboot a host and extract log files immediately after the host fails. If another host fails before you collect the diagnostic data of the first host, the second host does not save the core dump.

You can use the `vicfg-dumppart` command to query, set, and scan an ESX/ESXi host's diagnostic partitions. The *ESX Configuration Guide* and the *ESXi Configuration Guide* explain how to set up diagnostic partitions with the vSphere Client. The *Fibre Channel SAN Configuration Guide* and the *iSCSI SAN Configuration Guide* explain how to manage diagnostic partitions on a Fibre Channel or hardware iSCSI SAN.

Diagnostic partitions can include, in order of suitability, parallel adapter, block adapter, Fibre Channel, or hardware iSCSI partitions.

IMPORTANT When you list diagnostic partitions, software iSCSI partitions are included. However, VMware recommends that you not use software iSCSI partitions as diagnostic partitions.

The following example scenario changes the diagnostic partition.

To manage a diagnostic partition

1 Show the diagnostic partition the VMkernel uses.

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

2 Display information about all partitions that can be used as diagnostic partitions. Use `-l` for basic information, `-f` for detailed information.

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

The output might look as follows.

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

3 Deactivate the diagnostic partition.

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

The ESX/ESXi system is now without a diagnostic partition, and you must immediately set a new one.

4 Set the active partition to `naa.<naa_ID>.`

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

5 Run `vicfg-dumppart -t` again to verify that a diagnostic partition is set.

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

If a diagnostic partition is set, the command displays information about it. Otherwise, the command informs you that no partition is set.

Configuring Syslog on ESXi Hosts

All ESX/ESXi hosts run a syslog service (`syslogd`), which logs messages from the VMkernel and other system components to a file.

On an ESXi host, you can use the vSphere Client or the `vicfg-syslog` vCLI command to configure the following options:

- **Remote host.** Remote host to which `syslog` messages are forwarded. The remote host must have a syslog service installed and correctly configured to receive the forwarded syslog messages. See the documentation for the syslog service installed on your remote host for information on configuration.
- **Remote port.** Port on which the remote host receives syslog messages.

IMPORTANT You cannot use the vSphere Client or `vicfg-syslog` to configure syslog behavior for an ESX/ESXi host. You must edit the `/etc/syslog.conf` file to configure syslog for an ESX host.

You do not have to restart the syslog service after configuring the settings.

You cannot use `vicfg-syslog` to specify a datastore path to a file in which `syslogd` logs all messages. Use the vSphere Client to specify a datastore path. See the *Datacenter Administration Guide*.

To configure the syslog service

1 Run `vicfg-syslog --show` to display the syslog server configuration.

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

2 Run `vicfg-syslog --setserver` to set a remote server as the syslog server.

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

Makes `mysyslogserver` the syslog server for the ESX/ESXi host specified in `<conn_options>`.

3 Run `vicfg-syslog --setport` to set the port for the syslog server.

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

Managing ESX/ESXi SNMP Agents with vicfg-snmp

Simple Network Management Protocol (SNMP) allows management programs to monitor and control networked devices. vCenter Server and ESX/ESXi systems include different SNMP agents:

- The SNMP agent included with vCenter Server can send traps when the vCenter Server system is started or when an alarm is triggered on vCenter Server. The vCenter Server SNMP agent functions only as a trap emitter and does not support other SNMP operations such as `GET`.

  IMPORTANT You can manage the vCenter Server agent with the vSphere Client, but not with the vCLI command.

- ESX/ESXi 4.0 and later includes an SNMP agent embedded in the host daemon (`hostd`) that can send traps and receive polling requests such as `GET` requests.

Versions of ESX released before ESX/ESXi 4.0 included a Net-SNMP-based agent. You can continue to use this Net-SNMP-based agent in ESX 4.x with MIBs supplied by your hardware vendor and other third-party management applications. However, to use the VMware MIB files, you must use the embedded SNMP agent.

To use the NET-SNMP based agent and embedded SNMP agent at the same time, make one of the agents listen on a nondefault port. By default, both agents use the same port.

The embedded SNMP agent is disabled by default. Configuring and enabling the agent requires that you perform the following tasks:

1 Configure SNMP Communities. See "Configuring SNMP Communities" on page 127.

2 Configure the SNMP Agent. You have the following choices:

   - "Configuring the SNMP Agent to Send Traps" on page 128
   - "Configuring the SNMP Agent for Polling" on page 128

Configuring SNMP Communities

Before you enable the ESX/ESXi embedded SNMP agent, you must configure at least one community for the agent.

An SNMP community defines a group of devices and management systems. Only devices and management systems that are members of the same community can exchange SNMP messages. A device or management system can be a member of multiple communities.

To configure SNMP communities, run `vicfg-snmp -c`, specifying a comma-separated list of communities. For example:

```
vicfg-snmp <conn_options> -c public, internal
```

Each time you specify a community with this command, the settings that you specify overwrite the previous configuration.

Configuring the SNMP Agent to Send Traps

You can use the ESX/ESXi embedded SNMP agent to send virtual machine and environmental traps to management systems. To configure the agent to send traps, you must specify a target (receiver) address, the community, and an optional port. If you do not specify a port, the SNMP agent sends traps to UDP port 162 on the target management system by default.

To configure a trap destination

1 Make sure a community is set up.

```
vicfg-snmp <conn_options> --show

Current SNMP agent settings:
Enabled: 1
UDP port: 161
Communities: public
Notification targets:
```

2 Run `vicfg-snmp --target` with the target address, port number, and community.

```
vicfg-snmp <conn_options -t target.example.com@163/public
```

Each time you specify a target with this command, the settings you specify overwrite all previously specified settings. To specify multiple targets, separate them with a comma.

You can change the port that the SNMP agent sends data to on the target using the `--targets` option. That port is UDP 162 by default.

3 (Optional) Enable the SNMP agent if it is not yet running.

```
vicfg-snmp <conn_options> --enable
```

4 (Optional) Send a test trap to verify that the agent is configured correctly.

```
vicfg-snmp <conn_options> --test
```

The agent sends a `warmStart` trap to the configured target.

Configuring the SNMP Agent for Polling

If you configure the ESX/ESXi embedded SNMP agent for polling, it can listen for and respond to requests such as GET requests from SNMP management client systems.

By default, the embedded SNMP agent listens on UDP port 161 for polling requests from management systems. You can use the `vicfg-snmp` command to configure an alternative port. To avoid conflicts with other services, use a UDP port that is not defined in `/etc/services`.

IMPORTANT Both the embedded SNMP agent and the Net-SNMP-based agent available in the ESX service console listen on UDP port 161 by default. Change the port for one agent to enable both agents for polling.

To configure the SNMP agent for polling

1 (Optional) Specify a port for listening for polling requests.

```
vicfg-snmp <conn_options> -p <port>
```

2 (Optional) If the SNMP agent is not enabled, enable it.

```
vicfg-snmp <conn_options> --enable
```

3 Run `vicfg-snmp --test` to validate the configuration.

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
```

ESX, ESXi, and Virtual Machine Logs

Logs can help you find out what happened if commands do not have the desired results. You can find the following logs on your ESX/ESXi system.

Component	Location
ESX Server 2.x service log	`/var/log/vmware/vmware-serverd.log`
ESX Server 3.x or ESX service log	`/var/log/vmware/hostd.log`
vSphere client agent log	`/var/log/vmware/vpx/vpxa.log`
Virtual machine kernel core file	After you reboot your machine, files `/root/vmkernel-log.<date>` and `/root/vmkernel-core.<date>` are present.
Syslog log	`/var/log/messages`
Service console availability report	`/var/log/vmkernel`
VMkernel messages, alerts, and availability report	`/var/log/vmkernel`
VMkernel warning	`/var/log/vmkwarning`
Virtual machine log file	`vmware.log` in the same directory as the VMX file for the virtual machine
Virtual machine configuration file	`<virtual_machine_name>/<virtual_machine_name>.vmx` located on a datastore associated with the managed host. Use the virtual machine summary page in the vSphere Client to determine the datastore on which this file is located.

Enabling and Disabling CIM Providers

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.

An exception is the `vicfg-advcfg -s` option, which you can use to enable and disable OEM or Custom (IHV) CIM providers and VMW groups. With the VMware CIM APIs, developers can build standards-based CIM-compliant management applications to manage ESX/ESXi hosts. See the VMware CIM API documentation.

- Enable CIM providers as follows:

  OEM CIM providers `vicfg-advcfg <conn_options> -s 1 CIMOEMProvidersEnabled`

  IHV CIM providers `vicfg-advcfg <conn_options> -s 1 CIMCustomProvidersEnabled`

  All CIM providers `vicfg-advcfg <conn_options> -s 1 UserVars.CIMEnabled`

- Enable or disable VMW groups. By default, all groups are enabled. If you run `vicfg-advcfg -s` to disable a group, the change takes effect after SFCBD service restart or ESX/ESXi host reboot. For example, you can disable all CIM providers as follows:

  ```
  vicfg-advcfg <conn_options> -s 0 UserVars.CIMEnabled
  ```

  The following provider groups are available:

  - `UserVars.CIMEnabled`
  - `UserVars.CIMemulexProviderEnabled`
  - `UserVars.CIMlsiProviderEnabled`
  - `UserVars.CIMqlogicProviderEnabled`
  - `UserVars.CIMvmw_hdrProviderEnabled`
  - `UserVars.CIMvmw_kmoduleEnabled`
  - `UserVars.CIMvmw_lsiProviderEnabled`
  - `UserVars.CIMvmw_swmgtProviderEnabled`

vSphere CLI Command Overviews 12

The *vSphere CLI Reference*, available on the vSphere CLI documentation page, is a complete reference to all vCLI commands except `resxtop` and `esxcli`. This chapter gives an overview of all commands and points to related documentation.

This chapter includes the following topics:

List of Available Commands

Table 12-1 lists all vCLI commands in alphabetical order and points to the vCLI discussion in this document and related documentation.

Table 12-1. vCLI Commands Supported by ESX/ESXi

Command	Description	See
`esxcli`	Supports a variety of name spaces and commands listed in "esxcli Command Overview" on page 136.	"Managing Path Policies with esxcli" on page 55. "Setting Up Ports for iSCSI Multipathing" on page 80. "Managing Third-Party Storage Arrays with esxcli" on page 97. "Forcibly Stopping Virtual Machines" on page 95. `esxcli` supports other commands for working with storage, networking, and virtual machines.
`resxtop`	Monitors in real time how ESX/ESXi hosts use resources. Runs in interactive or batch mode. This command is supported only on Linux.	"Using resxtop for Performance Monitoring" on page 125. See the *Resource Management Guide* 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.	"Migrating Virtual Machines with svmotion" on page 59.
`vicfg-advcfg`	Performs advanced configuration including enabling and disabling CIM providers. Use this command as instructed by VMware.	"Enabling and Disabling CIM Providers" on page 129.
`vicfg-authconfig`	Allows you to remotely configure Active Directory settings for an ESX/ESXi host.	"Using vicfg-authconfig for Active Directory Configuration" on page 32.

Table 12-1. vCLI Commands Supported by ESX/ESXi (Continued)

Command	Description	See
`vicfg-cfgbackup`	Backs up the configuration data of an ESXi system and restores previously saved configuration data.	"Backing Up Configuration Information with vicfg-cfgbackup" on page 28. See the *ESXi Embedded and vCenter Server Setup Guide* for an in-depth discussion that includes step-by-step instructions.
`vicfg-dns`	Specifies an ESX/ESXi host's DNS (Domain Name Server) configuration.	"Setting the DNS Configuration" on page 119.
`vicfg-dumppart`	Manages diagnostic partitions.	"Managing Diagnostic Partitions with vicfg-dumppart" on page 125.
`vicfg-hostops`	Manages hosts.	"Stopping, Rebooting and Examining Hosts with vicfg-hostops" on page 27. "Entering and Exiting Maintenance Mode with vicfg-hostops" on page 28.
`vicfg-ipsec`	Sets up IPsec (Internet Protocol Security), which secures IP communications coming from and arriving at ESX/ESXi hosts. ESX/ESXi hosts support IPsec using IPv6.	"Using vicfg-ipsec for Secure Networking" on page 121.
`vicfg-iscsi`	Manages iSCSI storage.	"Managing iSCSI Storage" on page 65.
`vicfg-module`	Enables VMkernel options. Use this command with the options listed in this document, or as instructed by VMware.	"Managing VMkernel Modules with vicfg-module" on page 32.
`vicfg-mpath` `vicfg-mpath35`	Configures storage arrays. Use `vicfg-mpath35` for ESX/ESXi 3.5 hosts.	"Managing Paths with vicfg-mpath" on page 53.
`vicfg-nas`	Manages NAS file systems.	"Managing NFS/NAS Datastores with vicfg-nas" on page 58.
`vicfg-nics`	Manages the ESX/ESXi host's physical NICs.	"Managing Uplink Adapters with vicfg-nics" on page 116.
`vicfg-ntp`	Specifies the NTP (Network Time Protocol) server.	"Adding and Starting an NTP Server" on page 120.
`vicfg-rescan`	Rescans the storage configuration.	"Rescanning Storage Adapters with vicfg-rescan" on page 63.
`vicfg-route`	Manipulates the ESX/ESXi host's route entry.	"Managing the IP Gateway" on page 120.
`vicfg-scsidevs`	Finds available LUNs.	"Examining LUNs with vicfg-scsidevs" on page 52.
`vicfg-snmp`	Manages the Simple Network Management Protocol (SNMP) agent.	"Managing ESX/ESXi SNMP Agents with vicfg-snmp" on page 127. Using SNMP in a vSphere environment is discussed in detail in the *Datacenter Administration Guide*.
`vicfg-syslog`	Specifies the syslog server and the port to connect to that server for ESXi hosts.	"Configuring Syslog on ESXi Hosts" on page 126 The *Datacenter Administration Guide* explains how to set up system logs using the vSphere Client.
`vicfg-user`	Creates, modifies, deletes, and lists local direct access users and groups of users.	"Managing Users" on page 83 See the *Datacenter Administration Guide* for discussions of custom roles and of security implications of user management.
`vicfg-vmknic`	Adds, deletes, and modifies VMkernel network interfaces.	"Adding and Modifying VMkernel Network Interfaces with vicfg-vmknic" on page 117
`vicfg-volume`	Supports resignaturing a the copy of a VMFS volume and mounting and unmounting the copy.	"Managing Duplicate VMFS Datastores with vicfg-volume" on page 61.

Table 12-1. vCLI Commands Supported by ESX/ESXi (Continued)

Command	Description	See
`vicfg-vswitch`	Adds or removes virtual switches or modifies virtual switch settings.	"Setting Up Virtual Switches and Associating a Switch with a Network Interface" on page 114.
`vifs`	Performs file system operations such as retrieving and uploading files on the ESXi system.	"Managing the Virtual Machine File System with vmkfstools" on page 36
`vihostupdate` `vihostupdate35`	Manages updates of ESX/ESXi hosts. Use `vihostupdate35` for ESXi 3.5 hosts.	"Managing Host Updates with vihostupdate" on page 29. See also the *ESXi Upgrade Guide*.
`vmkfstools`	Creates and manipulates virtual disks, file systems, logical volumes, and physical storage devices on an ESX/ESXi host.	"Managing the Virtual Machine File System with vmkfstools" on page 36.
`vmware-cmd`	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.	"Managing Virtual Machines" on page 89.

Supported Platforms for Commands

vCLI 4.0 and later 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`
- `vicfg-ipsec`

You cannot run the `vihostupdate` and `vicfg-mpath` commands that are in a vCLI 4.x installation against ESX/ESXi 3.5 or vCenter 2.5 systems. Instead, run `vihostupdate35` and `vicfg-mpath35`, included in the vCLI 4.x installation, against those systems. `vihostupdate35` is supported for ESXi but not ESX.

IMPORTANT If you run vCLI 4.x 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 vCLI 4.x options are supported against hosts running ESX/ESXi 3.5 Update 2 or later even though they were not supported in RCLI version 3.5.

Run a vCLI 4.x command with `--help` for information on option support with ESX/ESXi 3.5 Update 2, or see the VMware knowledge base article at http://kb.vmware.com/kb/1008940 for more detail.

Table 12-2 lists platform support for the different vCLI 4.x commands. These commands have not been tested against VirtualCenter 2.5 Update 2 systems. You can, however, connect to a vCenter Server 4.x system and target ESX/ESXi 3.5 Update 2 hosts.

Table 12-2. Platform Support for vCLI 4.x Commands

Command	ESXi 4.x	ESX 4.x	VC 4.x	ESXi 3.5 U2+	ESX 3.5 U2+
esxcli	Yes	Yes	No	No	No
resxtop	Yes	Yes	Yes	Yes	Yes
svmotion	No	No	Yes	No	No
vicfg-advcfg	Yes	Yes	Yes	Yes	Yes
vicfg-authconfig	Yes	Yes	Yes	No	No
vicfg-cfgbackup	Yes	No	No	Yes	No
vicfg-dns	Yes	Yes	Yes	Yes	Yes
vicfg-dumppart	Yes	Yes	Yes	Yes	Yes
vicfg-hostops	Yes	Yes	Yes	No	No
vicfg-ipsec	Yes	Yes	No	No	No
vicfg-iscsi	Yes	Yes	Yes	No	No
vicfg-module	Yes	Yes	Yes	Yes	Yes
vicfg-mpath	Yes	Yes	Yes	Use vicfg-mpath35 instead.	
vicfg-nas	Yes	Yes	Yes	Yes	Yes
vicfg-nics	Yes	Yes	Yes	Yes	Yes
vicfg-ntp	Yes	Yes	Yes	Yes	Yes
vicfg-rescan	Yes	Yes	Yes	Yes	Yes
vicfg-route	Yes	Yes	Yes	Yes	Yes
vicfg-scsidevs	Yes	Yes	Yes	No	No
vicfg-snmp	Yes	Yes	No	Yes	Yes
vicfg-syslog	Yes	No	Yes	Yes	No
vicfg-user	Yes	Yes	No	Yes	Yes
vicfg-vmhbadevs	Not included in vCLI 4.x. Use vicfg-scsidevs instead.			Yes	Yes
vicfg-vmknic	Yes	Yes	Yes	Yes	Yes
vicfg-volume	Yes	Yes	Yes	No	No
vicfg-vswitch	Yes	Yes	Yes	Yes	Yes
vifs	Yes	Yes	No	Yes	Yes
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 12-3 lists platform support for the different vCLI 3.5 commands. These commands are not supported against vSphere 4.x systems.

Table 12-3. Platform Support for vCLI 3.5 Commands

Command	ESXi 3.5 U2+	ESX 3.5 U2+	VC 2.5 U2+
esxcli	No	No	No
resxtop	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-vmknic	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 vCLI 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 vCLI commands.

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

Table 12-4 lists all vCLI commands for which a command with an `esxcfg` prefix is available.

Table 12-4. Commands with an `esxcfg` Prefix

Command with vicfg prefix	Command with esxcfg prefix
vicfg-advcfg	esxcfg-advcfg
vicfg-cfgbackup	esxcfg-cfgbackup
vicfg-dns	esxcfg-dns

Table 12-4. Commands with an `esxcfg` Prefix (Continued)

Command with vicfg prefix	Command with esxcfg prefix
`vicfg-dumppart`	`esxcfg-dumppart`
`vicfg-module`	`esxcfg-module`
`vicfg-mpath`	`esxcfg-mpath`
`vicfg-nas`	`esxcfg-nas`
`vicfg-nics`	`esxcfg-nics`
`vicfg-ntp`	`esxcfg-ntp`
`vicfg-rescan`	`esxcfg-rescan`
`vicfg-route`	`esxcfg-route`
`vicfg-scsidevs`	`esxcfg-scsidevs`
`vicfg-snmp`	`esxcfg-snmp`
`vicfg-syslog`	`esxcfg-syslog`
`vicfg-vmknic`	`esxcfg-vmknic`
`vicfg-volume`	`esxcfg-volume`
`vicfg-vswitch`	`esxcfg-vswitch`

esxcli Command Overview

The `esxcli` vCLI command differs from other vCLI commands. The command is not a Perl script and you cannot invoke it 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.

The command is installed in the same location as other vCLI commands and you invoke it with the same connection options as other vCLI commands. See "Running vCLI Commands" on page 19.

The command has the following syntax:

```
esxcli <conn_options> <namespace> <app> <cmd> [cmd options]
```

Option	Description
`<conn_options>`	Connection parameters for the vCLI 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 "vCLI Connection Options" on page 23. `ESXCLI` does not support the credential store.
`<namespace>`	Namespace. One of the following: `nmp` – VMware native multipathing commands. `swiscsi` – Commands in the software iSCSI and dependent hardware iSCSI namespace. `corestorage` – VMware core storage commands.
`<app>`	Area within the namespace to which the command applies.
`<cmd>`	Command to be called.
`<cmd options>`	Command options.

Help for esxcli

Command-line help for the `esxcli` vCLI is available on a per-level basis.

IMPORTANT When using the `esxcli` vCLI, you must supply connection information including a user name and password, even if you call the command with `--help`. `esxcli` displays the information about available commands and options on the server you specify.

This behavior differs from `--help` for other vCLI commands because other commands do not change depending on the target server.

How you call help depends on what you call help for.

Command	Example	Output
`esxcli`	`esxcli <Enter>`	Lists all supported name spaces on this system.
`esxcli --help` `esxcli -?`	`esxcli --help` `esxcli -?`	Displays help for supported connection options.
`esxcli <conn_options> --help` `esxcli <conn_options> -?`	`esxcli --server S1 --help` `esxcli --server S1 -?`	Displays help for supported name spaces.
`esxcli <conn_parms> <namespace> --help` `esxcli <conn_parms> <namespace> -?`	`esxcli --server S1 nmp --help` `esxcli --server S1 nmp -?`	Displays help for supported apps for this name space.
`esxcli <conn_options> <namespace> <app> --help` `esxcli <conn_options> <namespace> <app> -?`	`esxcli --server S1 nmp device --help` `esxcli --server S1 nmp device -?`	Displays help for supported commands for this app.
`esxcli <conn_options> <namespace> <app> <command> --help` `esxcli <conn_options> <namespace> <app> <command> -?`	`esxcli --server S1 nmp device setpolicy --help` `esxcli --server S1 nmp device setpolicy -?`	Displays help for supported options for this command.

This section is a reference to `esxcli` commands, organized on a per-namespace basis. The reference points to places in this manual that discuss using the command.

esxcli corestorage Namespace

The `esxcli corestorage claiming` commands include pluggable storage direct path claiming commands and claim rule commands. These commands operate on the rules that determine which PSA plugin is used to claim storage paths.

claiming Commands

The `esxcli corestorage claiming` commands apply to the pluggable storage direct path claiming system. These operations allow a user to directly control the claiming and unclaiming process. The results of these operations are temporary. Claiming operations that need to persist after a reboot should use claim rules instead (see "claimrule Commands" on page 138).

You can specify the following claiming commands.

Command	Description
`autoclaim`	Do not use this command unless instructed to do so by VMware support staff.
`reclaim`	Attempts to unclaim all paths to a device and runs the loaded claim rules on each of the unclaimed paths to reclaim them. Supports the following options: ■ `-d|--device` Name of a device on which all paths will be unclaimed and then reclaimed. ■ `-h|--help` Show the help message.

Command	Description
`unclaim`	Unclaims a path or set of paths, disassociating it from a PSA plugin. It is normal for path claiming to fail especially when unclaiming by plugin or adapter. Only inactive paths with no I/O can be unclaimed. Typically the ESXi USB partition and devices with VMFS volumes on them are not unclaimable. Unclaiming does not persist and periodic path claiming reclaims unclaimed paths a short time after the unclaim operations unless claim rules are configured to mask the path. See "Masking Paths with esxcli corestorage claimrule" on page 57. See "esxcli corestorage claiming unclaim" on page 105 for a list of options.

claimrule Commands

The `esxcli corestorage claimrule` commands operate on the rules used to determine which PSA plugin is used to claim storage paths. You can also use these commands to mask paths, see "Masking Paths with esxcli corestorage claimrule" on page 57.

The following commands are supported.

Command	Description
`add`	Adds a claim rule to the set of claim rules on the system. Supports options listed in "Adding Claim Rules with esxcli corestorage claimrule add" on page 106.
`convert`	Converts ESX 3.x style `/adv/Disk/MaskLUNs` LUN masks to claim rule format. This conversion does not work for all input MaskLUNs variations. After the convert operation has generated a list of claim rules, inspect those rules first. If the suggested LUN mask claim rules are correct, use the `--commit` option to write the list to the configuration file. See "Converting ESX 3.5 LUN Masks to Claim Rule Format" on page 107.
`delete`	Deletes a claim rule. See "Deleting Claim Rules with esxcli corestorage claimrule delete" on page 108.
`list`	Lists all claim rules on the system.
`load`	Loads path claiming rules from the configuration file into the VMkernel. See "Loading Claim Rules with esxcli corestorage claimrule load" on page 108.
`move`	Moves a claim rule from one rule id to another. See "Moving Claim Rules with esxcli corestorage claimrule move" on page 109.
`run`	Runs path claiming rules. See "esxcli corestorage claimrule run" on page 109.

device Commands

The `esxcli corestorage device list` command applies to the pluggable storage architecture's logical devices on the system. The command is used in conjunction with hardware acceleration, discussed in the *Fibre Channel SAN Configuration Guide* and the *iSCSI SAN Configuration Guide*.

Command	Description
`list`	For devices currently registered with the PSA, lists the attached filters. Supports the following options: ■ `-d|--device` Filters the output of this command to only show a single device. ■ `-h|--help` Shows the help message.

plugin commands

The `esxcli corestorage plugin list` command works on PSA plugins. The command is used in conjunction with hardware acceleration, discussed in the *Fibre Channel SAN Configuration Guide* and the *iSCSI SAN Configuration Guide*.

Command	Description
`list`	Lists PSA plugins available on the specified system. Supports the following options: ■ `–N|––plugin–class=<str>` Indicates the class of plugin to limit the list to. The following values are supported: ■ `Filter`: Filter plugins ■ `MP`: Multipathing plugins ■ `VAAI`: VAAI plugins ■ `all`: All PSA Plugins (default) For example, you can list all PSA plugins that are multipathing plugins by running the following command: `esxcli corestorage plugin list ––plugin–class=MP`

esxcli network Namespace

The `esxcli network` namespace allows you to list information about the current network status. The namespace supports the `connections` and `neighbors` apps.

connections list Command

The `esxcli network connections list` command lists active TCP/IP connections.

Command	Description
`list`	Lists active TCP/IP connections. Supports the following options: ■ `–h|––help` Shows the help message. ■ `–t | ––type` Specifies the connection type for which you want to list active connections. Supported types are `ip`, `tcp`, `udp`, and `all`.

neighbors show Command

The `esxcli network neighbors show` command lists active ARP table entries.

Command	Description
`show`	Lists active ARP table entries. Supports the following options: ■ -h|--help Shows the help message. ■ -v | --version IP version to list entries for. Supported types are 4, 6, and all.

esxcli nmp Namespace

The `esxcli nmp` namespace allows you to view and manipulate the Native Multipathing Plugin. See the *ESX Configuration Guide* for background information about the Pluggable Storage Array architecture and the Native Multipathing Plugin.

boot restore Command

The `esxcli nmp boot restore` command can be used to restore configuration state of the NMP plugin at boot time.

```
esxcli nmp boot restore
```

The command supports a `––help` option but no other options.

device Commands

The `esxcli nmp device` commands can be used for inspecting and managing the devices that are currently claimed by the VMware NMP. The following commands are supported.

Command	Description
`list`	Lists the devices currently controlled by VMware NMP and shows the SATP and PSP information associated with those devices. ■ `-d|--device` Filters the output of this command to only show a single device. ■ `-h|--help` Shows the help message.
`setpolicy`	Allows setting of the PSP for the given device to one of the policies loaded on the system. ■ `-E|--default` Sets the PSP for the assigned SATP for this device back to the default. ■ `-d|--device` Device you wish to set the PSP for. ■ `-h|--help` Shows the help message. ■ `-P|--psp` PSP you wish to assign to the device specified by `--device`.

fixed Commands

The `esxcli nmp fixed` commands apply to the Fixed PSP. The following commands are supported.

Command	Description
`getpreferred`	Retrieves fixed PSP settings for the specified device. ■ `-d|--device` Device you wish to get the preferred path for. ■ `-h|--help` Shows the help message.
`setpreferred`	Sets the preferred path on a specified device controlled by the fixed path selection policy. ■ `-E|--default` Clears the preferred path selection for the given device. ■ `-d|--device` Device you wish to set the preferred path for. This device must be controlled by the `fixed` PSP. ■ `-h|--help` Shows the help message. ■ `-p|--path` Path you wish to set as the preferred path for the given device.

path Commands

The `esxcli nmp path list` command lists the paths that are currently claimed by VMware NMP.

Command	Description
`list`	List the paths currently claimed by VMware NMP and show the SATP and PSP information associated with that path. ■ `-d|--device` Filters the output of this command to only show paths to a single device. ■ `-h|--help` Shows the help message. ■ `-p|--path` Filters the output of this command to only show a single path

psp Commands

The `esxcli nmp psp` commands apply to the PSP for the NMP. The following commands are supported.

Command	Description
`getconfig`	Retrieves per-path or per-device PSP configuration parameters. ■ `-d|--device` Device you wish to get PSP configuration for. If you specify `--device`, you cannot specify `--path`. ■ `-h|--help` Show the help message. ■ `-p|--path` Path you wish to get PSP configuration for. If you specify `--path`, you cannot specify `--device`.
`list`	Lists the PSP that are currently loaded into the NMP system and display information about those PSPs.

Command	Description
`setconfig`	Allows setting of per-path or per-device PSP configuration parameters. Sets the configuration for the given device or path with the PSP it is currently configured with. ■ `-c|--config` Configuration string you wish to set for the given path or device. ■ `-d|--device` Device you wish to set PSP configuration for. If you specify `--device`, you cannot specify `--path`. ■ `-h|--help` Show the help message. ■ `-p|--path` Path you wish to set PSP configuration for. If you specify `--path`, you cannot specify `--device`.

roundrobin Commands

`esxcli nmp roundrobin` commands apply to the round robin PSP. The following commands are supported:

Command	Description
`getconfig`	Allows retrieval of round robin PSP settings for a given device. ■ `-d|--device` Device you wish to get the settings for. ■ `-h|--help` Shows the help message.
`setconfig`	Allows setting of the round robin path options on a given device controlled by the Round Robin PSP. See "Customizing Round Robin Setup with esxcli nmp roundrobin" on page 101.

satp Commands

`esxcli nmp satp` commands apply to the SATPs for the VMware NMP. The following commands are supported:

Command	Description
`addrule`	Adds a rule to the list of claim rules for the given SATP. See "Adding SATP Rules" on page 102
`deleterule`	Deletes a rule from the list of claim rules for the given SATP. See "Deleting SATP Rules" on page 103
`getconfig`	Allows retrieval of per-path or per-device SATP configuration parameters.
`list`	Lists the SATPs that are currently loaded into the NMP system and display information about those SATPs. See "Retrieving Information About SATPs" on page 102.
`listrules`	Lists the claiming rules for SATPs.
`setconfig`	Allows setting of per-path or per-device SATP configuration parameters. See "esxcli nmp roundrobin setconfig" on page 101.
`setdefaultpsp`	Sets the default PSP for a given SATP. See "Setting the Default PSP" on page 104.

esxcli swiscsi Namespace

The commands in the `swiscsi` namespace support software iSCSI management.

nic Commands

`esxcli swiscsi nic` commands allow you to perform operations on iSCSI network interfaces. You can bind an existing VMkernel network interface for use by iSCSI, remove a VMkernel network interface from the current iSCSI configuration, and list network port bindings. The following commands are supported.

Command	Description
`add`	Binds an existing VMkernel network interface that is not used for other purposes to the current software iSCSI configuration. Supports the following options: ■ `-d | --adapter` Name of the software iSCSI adapter to bind to the VMkernel network interface. ■ `-h | --help` Shows the help message. ■ `-n | --nic` Name of the VMkernel network interface to bind as an iSCSI network interface. See "Adding and Modifying VMkernel Network Interfaces with vicfg-vmknic" on page 117 for information on creating a new VMkernel network interface for iSCSI.

Command	Description
`list`	Lists network port bindings for the VMkernel network interface. Supports the following options: ■ `-d | --adapter` Name of the iSCSI adapter for which you want to list network port bindings. ■ `-h | --help` Shows the help message.
`remove`	Removes a VMkernel network interface from the current software iSCSI configuration. Supports the following options: ■ `-d | --adapter` Name of the software iSCSI adapter from which you want to remove the network interface. ■ `-h | --help` Shows the help message. ■ `-n | --nic` Name of the VMkernel network interface to remove the binding from.

session Commands

`esxcli swiscsi session` commands allow you to add, list, and remove iSCSI login sessions. See "Managing iSCSI Sessions" on page 80. The following commands are supported.

Command	Description
`add`	Adds a login session to the current iSCSI configuration. Supports the following options: ■ `-d | --adapter` Name of the software iSCSI adapter you want to add a session to. ■ `-h | --help` Shows the help message. ■ `-s | --isid` iSCSI ID of the session you want to duplicate for login. Run `esxcli swiscsi session list` for that device to see the session iSCSI ID (`session_isid`). ■ `-t | --target` Name of the target to log in to.
`list`	Lists information about sessions for a specified adapter or target. You need this information when you want to remove a session from an adapter. Supports the following options: ■ `-d | --adapter` Name of the adapter you want to list information for. ■ `-h | --help` Shows the help message. ■ `-t | --target` Name of the target to list information for.
`remove`	Removes the specified iSCSI session or sessions from the iSCSI configuration. Supports the following options: ■ `-d | --adapter` Name of the software iSCSI adapter you want to remove a session from. ■ `-h | --help` Shows the help message. ■ `-s | --isid` The iSCSI ID of the session you want to remove. Run `esxcli swiscsi session list` for that device to see the target iSCSI ID (`session_isid`) ■ `-t | --target` Name of the target to remove. Run `esxcli swiscsi session list` for that device to see the target.

vmknic Commands

The `esxcli swiscsi vmknic list` command allows you to list VMkernel network interfaces available for binding to a particular iSCSI adapter.

Command	Description
`list`	Lists all VMkernel network interfaces. Supports the following options: `-d|--adapter` The iSCSI adapter name (vmhba#) for which you want to list VMkernel network interface. `-h|--help` Show the help message.

vmnic Commands

The `esxcli swiscsi vmnic list` command allows you to list available uplink adapters for use with a specified iSCSI adapter.

Command	Description
`list`	`-d|--adapter` The iSCSI adapter name (vmhba#). `-h|--help` Shows the help message.

esxcli vaai Namespace

The vaai (vStorage APIs for Array Integration) namespace includes commands that allow storage management on a per-virtual machine or per-VMDK file basis.

If you add a third-party VAAI plug-in to your environment, a reboot is required in most cases. See your storage vendor's documentation for more information.

device list Command

The `esxcli vaai device list` command can run on devices that have the VMware VAAI filter attached.

Command	Description
`list`	Lists the devices claimed by the VMware VAAI Filter Plugin and shows the VAAI Filter plugin information associated with that device. Supports the following options: ■ `–d|––device` Filters the output of this command to only show a single device. ■ `–h|––help` Shows the help message.

esxcli vms Namespace

The `vms` (virtual machines) namespace supports virtual machine operations. It currently allows you to forcibly stop a virtual machine, as discussed in "Forcibly Stopping Virtual Machines" on page 95.

vm Commands

Command	Description

`kill`	Allows you to forcibly stop virtual machines that do not respond to normal stop operations. Supports the following options:
	■ -t|--type Type of kill operation, Use the types sequentially (soft before hard, hard before force). The following types are supported through the --type option:
	  ■ `soft` – Gives the VMX process a chance to shut down cleanly (like `kill S––SIGTERM`)
	  ■ `hard` – Stops the VMX process immediately (like `kill –9` or `kill –SIGKILL`)
	  ■ `force` – Stops the VMX process when other options do not work.
	  If all three options do not stop the virtual machine, reboot the ESX/ESXi system.
	■ `–w|––world–id` World ID of the virtual machine to kill. Run `esxcli vms vm list` to find the World ID.

`list`	Lists information about all running virtual machines, including the World ID of the virtual machine, needed by the `kill` command.

Index

P

R

S

W

Z