

Remote Command-Line Interface Installation and Reference Guide

Update 2 and later for
ESX Server 3.5, ESX Server 3i version 3.5, VirtualCenter 2.5

Remote Command-Line Interface Installation and Reference Guide

Revision: 20090313

Item: EN-000082-01

You can find the most up-to-date technical documentation on our 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

© 2007–2009 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

VMware, the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.

3401 Hillview Ave.

Palo Alto, CA 94304

www.vmware.com

Contents

About This Book	5
Product Naming Conventions	5
1 Remote CLI Installation and Execution	9
List of Available Commands	10
Using the VMware Remote CLI	12
Installing and Executing Remote CLI Commands on Linux	13
Unpacking and Installing the Remote CLI Package	13
Executing Remote CLI Commands	14
Uninstalling the Remote CLI Package	15
Installing and Executing Remote CLI Commands on Windows	15
Installing the Remote CLI Package	15
Executing Remote CLI Commands	16
Uninstalling the Remote CLI Package	16
Installing and Running the Remote CLI Virtual Appliance	17
Importing the Virtual Appliance	17
Running the Virtual Appliance	18
Specifying Options for Remote CLI Commands	19
Using a Session File	19
Setting Environment Variables	20
Using a Configuration File	20
Passing Parameters at the Command Line	21
Using Microsoft Windows Security Support Provider Interface (SSPI)	21
Available Options for Remote CLI Execution	22
Examples for Specifying Connection Options	24
Using Remote CLI Commands in Scripts	25
Example: Editing Files on the ESX Server Host	26
Example: Adding a NAS Datastore to Multiple ESX Server Hosts	26
2 Remote CLI Reference	27
Storage Management Commands	28
Managing Diagnostic Partitions with vicfg-dumppart	28
Configuring Storage Arrays with vicfg-mpath	30

- Managing NAS File Systems with vicfg-nas 33
- Rescanning with vicfg-rescan 34
- Finding Available LUNs with vicfg-vmhbadevs 35
- Networking Commands 36
 - Specifying the DNS Configuration with vicfg-dns 36
 - Managing Physical NICs with vicfg-nics 38
 - Specifying the NTP Server with vicfg-ntp 39
 - Manipulating the route Entry with vicfg-route 40
 - Managing VMkernel NICs with vicfg-vmknic 41
 - Managing Virtual Switches with vicfg-vswitch 42
- System Management Commands 45
 - Using resxtop for Performance Monitoring 45
 - Using vicfg-cfgbackup for Configuration Information 45
 - Setting Up SNMP with vicfg-snmp 47
 - Specifying the syslog Server with vicfg-syslog 49
 - Performing Maintenance with vihostupdate 50
- Virtual Machine Management Commands 51
 - Migrating Virtual Machines Using svmotion 52
- Miscellaneous Management Commands 54
 - Using vicfg-advcfg in Special Circumstances 55
 - Enabling VMkernel Module Options with vicfg-module 55
 - Managing Users and Groups with vicfg-user 56
- Performing File System Operations with vifs 58
- Commands with an esxcfg Prefix 61

- 3 Managing Storage with vmkfstools 63**
 - vmkfstools Command Syntax 64
 - Specifying Options 64
 - Specifying the Target 66
 - vmkfstools File System Options 67
 - vmkfstools Virtual Disk Options 69

- 4 Managing Virtual Machines with vmware-cmd 77**
 - Connection Options for vmware-cmd 78
 - General Options for vmware-cmd 78
 - Server Options for vmware-cmd 79
 - Virtual Machine Options for vmware-cmd 79
 - Values for powerop_mode 81
 - vmware-cmd Examples 83

- Index 85**

About This Book

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

- [Chapter 1](#) explains how to install Remote CLI packages, how to import the Remote CLI appliance, and how to execute Remote CLI commands on Windows or Linux. The chapter also includes a reference to connection options and other options you can use with each of the Remote CLI commands.
- [Chapter 2](#) provides reference documentation for most of the Remote CLI commands, organized by subject area. For each command, the chapter lists and discusses each option and gives examples.
- [Chapter 3](#) and [Chapter 4](#) provide reference documentation for two of the more complex Remote CLI commands.
 - [Chapter 3](#) discusses `vmkfstools`. It includes information about the command syntax, file system options, and virtual disk options.
 - [Chapter 4](#) discusses `vmware-cmd`. It includes information about connection options, general options, server options, and virtual machine options.

Product Naming Conventions

This book covers both ESX Server 3.5 and ESX Server 3i version 3.5. For ease of discussion, this book uses the following product naming conventions:

- For topics specific to ESX Server 3.5, this book uses the term “ESX Server 3.”
- For topics specific to ESX Server 3i version 3.5, this book uses the term “ESX Server 3i.”

- For topics common to both products, this book uses the term “ESX Server.”
- When the identification of a specific release is important to a discussion, this book refers to the product by its full, versioned name.
- When a discussion applies to all versions of ESX Server for VMware Infrastructure 3, this book uses the term “ESX Server 3.x.”

Intended Audience

This book is intended for administrators who want to use command-line interfaces instead of the VMware Infrastructure Client (VI Client) to manage and monitor the VMware Infrastructure environment. The information in this book is written for experienced Windows or Linux system administrators who are familiar with virtual machine technology and datacenter operations and know how to use commands in scripts.

Document Feedback

VMware welcomes your suggestions for improving our documentation. If you have comments, send your feedback to:

docfeedback@vmware.com

Related Documentation

The VMware Infrastructure documentation consists of the combined VMware VirtualCenter and ESX Server documentation set. The documentation set explains a lot of the background information for the commands discussed in this document, for example, storage, networking, and so on.

The VI Perl Toolkit documentation explains how you can use the VI Perl Toolkit and related utility applications to manage your VMware Infrastructure environment. You can use the Remote CLI commands in conjunction with the VI Perl Toolkit.

Technical Support and Education Resources

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

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

Online and Telephone Support

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.html

Support Offerings

Find out how VMware support offerings can help meet your business needs. Go to:

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

VMware Education Services

VMware courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. For more information about VMware Education Services, go to:

<http://mylearn1.vmware.com/mgrreg/index.cfm>

Remote CLI Installation and Execution

1

This chapter explains how to install and execute Remote CLI commands and includes a list of all supported commands that points to where each command is discussed.

Remote CLI commands allow you to perform many of the operations you might currently perform using the ESX Server 3 service console. Remote CLI commands are especially useful for an ESX Server 3i host because it does not include a service console. You can use Remote CLI commands in scripts that run on ESX Server 3i and ESX Server 3 version 3.5 hosts. You can also use Remote CLI commands for console access to ESX Server hosts.

Remote CLI commands run on top of the VMware Infrastructure Perl Toolkit (VI Perl Toolkit), which depends on Perl and a number of other libraries. Prerequisite software is included with the installation package on Windows and on supported Linux distributions. You can use an installable package or the Remote CLI appliance to run Remote CLI and VI Perl Toolkit commands.

This chapter includes the following topics:

- [“List of Available Commands”](#) on page 10
- [“Using the VMware Remote CLI”](#) on page 12
- [“Installing and Executing Remote CLI Commands on Linux”](#) on page 13
- [“Installing and Executing Remote CLI Commands on Windows”](#) on page 15
- [“Installing and Running the Remote CLI Virtual Appliance”](#) on page 17
- [“Specifying Options for Remote CLI Commands”](#) on page 19
- [“Available Options for Remote CLI Execution”](#) on page 22
- [“Using Remote CLI Commands in Scripts”](#) on page 25

List of Available Commands

[Table 1-1](#) lists all Remote CLI commands in alphabetical order and points to the Remote CLI discussion in this document.

Table 1-1. Remote CLI commands supported by ESX Server 3

Command	Description	See
resxtop	Monitors in real time how ESX Server hosts use resources. Runs in interactive or batch mode.	See “Using resxtop for Performance Monitoring” on page 45 for an introduction. See the <i>Resource Management Guide</i> for a detailed reference.
svmotion	Moves a virtual machine’s configuration file and optionally its disks while the virtual machine is running. Runs on a VirtualCenter Server.	See “Migrating Virtual Machines Using svmotion” on page 52 for an introduction. See the <i>Basic System Administration</i> guide for an in-depth discussion that includes examples.
vicfg-advcfg	Performs advanced configuration. Use this command as instructed by VMware.	See “Using vicfg-advcfg in Special Circumstances” on page 55.
vicfg-cfgbackup	Backs up the configuration data of an ESX Server 3i system and restores previously saved configuration data.	See “Using vicfg-cfgbackup for Configuration Information” on page 45. See the <i>ESX Server 3i Embedded Setup Guide</i> for an in-depth discussion that includes step-by-step instructions.
vicfg-dns	Specifies an ESX Server host’s DNS (Domain Name System) configuration.	See “Specifying the DNS Configuration with vicfg-dns” on page 36.
vicfg-dumppart	Manages diagnostic partitions.	See “Managing Diagnostic Partitions with vicfg-dumppart” on page 28.
vicfg-module	Enables VMkernel options. Use this command with the options listed in this document, or instructed by VMware.	See “Enabling VMkernel Module Options with vicfg-module” on page 55
vicfg-mpath	Configures storage arrays.	See “Configuring Storage Arrays with vicfg-mpath” on page 30.
vicfg-nas	Manages NAS file systems.	See “Managing NAS File Systems with vicfg-nas” on page 33.
vicfg-nics	Manages the ESX Server host’s physical NICs.	See “Managing Physical NICs with vicfg-nics” on page 38.
vicfg-ntp	Specifies the NTP (Network Time Protocol) server.	See “Specifying the NTP Server with vicfg-ntp” on page 39.
vicfg-rescan	Rescans the storage configuration.	See “Rescanning with vicfg-rescan” on page 34.

Table 1-1. Remote CLI commands supported by ESX Server 3 (Continued)

Command	Description	See
vicfg-route	Manipulates the ESX Server host's route entry.	See "Manipulating the route Entry with vicfg-route" on page 40.
vicfg-snmp	Sets up SNMP (Simple Network Management Protocol).	See "Setting Up SNMP with vicfg-snmp" on page 47. Using SNMP in a VMware Infrastructure environment is discussed in detail in the <i>Basic System Administration</i> manual.
vicfg-syslog	Specifies the syslog server and the port to connect to that server for ESX Server 3i hosts.	See "Specifying the syslog Server with vicfg-syslog" on page 49. The <i>Basic System Administration</i> document discusses system logs in more detail and explains how to set them up using the VI Client.
vicfg-user	Creates, modifies, deletes, and lists local direct access users and groups of users.	See "Managing Users and Groups with vicfg-user" on page 56.
vicfg-vmhbadevs	Finds available LUNs on the ESX Server host.	See "Finding Available LUNs with vicfg-vmhbadevs" on page 35.
vicfg-vmknic	Adds, deletes, and modifies virtual network adapters (VMkernel NICs).	See "Managing VMkernel NICs with vicfg-vmknic" on page 41.
vicfg-vswitch	Adds or removes virtual switches or modifies virtual switch settings.	See "Managing Virtual Switches with vicfg-vswitch" on page 42.
vifs	Performs file system operations such as retrieving and uploading files on the remote server.	See "Performing File System Operations with vifs" on page 58.
vihostupdate	Performs maintenance on ESX Server 3i hosts.	See "Performing Maintenance with vihostupdate" on page 50. Discussed in detail in the <i>ESX Server 3i Embedded Setup Guide</i> .
vmkfstools	Creates and manipulates virtual disks, file systems, logical volumes, and physical storage devices on an ESX Server host.	See "Managing Storage with vmkfstools" on page 63.
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.	See "Managing Virtual Machines with vmware-cmd" on page 77.

Using the VMware Remote CLI

You can install a Remote CLI package on either Linux or Microsoft Windows, or import a virtual appliance that includes the Remote CLI and all prerequisite software into your ESX Server host.

- **Remote CLI packages** – You can install a Remote CLI package on one server, which could become the administration server for all ESX Server hosts. See [“Installing and Executing Remote CLI Commands on Linux”](#) on page 13 and [“Installing and Executing Remote CLI Commands on Windows”](#) on page 15.

After you have installed the package, which includes the VI Perl Toolkit, you can run Remote CLI commands or invoke scripts from the operating system command line. Each time you run a command, you specify the connection parameters directly or indirectly. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

- **Remote CLI virtual appliance** – Download the Remote CLI virtual appliance and import it into a VirtualCenter Server or ESX Server host. The Remote CLI virtual appliance is a virtual machine prepackaged with a pared-down Linux operating system, the VI Perl Toolkit, and all Remote CLI commands. When the Remote CLI virtual appliance is available on your ESX Server host, you can run Remote CLI commands from the Linux console of the virtual appliance. See [“Installing and Running the Remote CLI Virtual Appliance”](#) on page 17.

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

- Open or access a Linux console on the virtual appliance, and enter Remote CLI commands there.
- Open a command prompt on a Linux or Windows system where you installed the Remote CLI and enter commands there.
- Prepare scripts with sets of Remote CLI commands. Then run the scripts from an administration server that has the Remote CLI package installed or from a Remote CLI virtual appliance Linux console. See [“Using Remote CLI Commands in Scripts”](#) on page 25.

When you execute commands, you must specify the ESX Server host on which you want to run the command and the user name and password, as discussed in “Specifying Options for Remote CLI Commands” on page 19.



CAUTION Specifying the password in plain text risks exposing the password to other users. The password might also become exposed in backup files. Provide a plain-text password only on client systems that you consider secure. Providing plain-text passwords on production systems is not recommended. You have two choices:

- If you use a Remote CLI interactively and do not specify a user name and password, you are prompted. What you type is not echoed to the screen.
- For noninteractive use, you can create a session file using the `save_session.pl` script included in the `apps/session` directory of the VI Perl Toolkit. See “Using a Session File” on page 19.

Installing and Executing Remote CLI Commands on Linux

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

- Red Hat Enterprise Linux (RHEL) 5.1
- Ubuntu Desktop 7.10
- Remote CLI virtual appliance

A number of prerequisite libraries are included in the default installation of each supported Linux distribution.

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

Unpacking and Installing the Remote CLI Package

To download and unpack the Remote CLI Package

- 1 Download the installer package from:
<http://www.vmware.com/go/remotecli>.
- 2 Open a shell prompt and navigate to the directory to which you downloaded the package.
- 3 Unpack the downloaded package.

To install the Remote CLI Package

- 1 Start the installer (`vmware-install-rcli.pl`).
The installer prompts you to accept the terms of the license agreement.
- 2 Type **yes** at the prompt to accept the license terms and press Enter to continue. If you do not type **yes** in full and press Enter, the installer cannot continue.
The installer prompts you to provide an installation location or to accept the default, which is `/usr/bin`.
- 3 Specify an installation directory, or press Enter to accept the default.
The installer checks whether the correct version of the VI Perl Toolkit is installed on your system and offers to replace it.

When the installation process completes:

- A success message appears.
- The installer lists different version numbers for required modules (if any).
- The prompt returns to the shell prompt.
- You can execute a Remote CLI command to verify that installation was successful. See [“Executing Remote CLI Commands”](#) on page 14.

A number of VI Perl utility applications and sample scripts that perform common operations on ESX Server hosts and virtual machines are included with the installation. See the VI Perl Toolkit documentation for a reference to all utility applications.

Executing Remote CLI Commands

After you have installed the Remote CLI, you can execute commands from the Linux command prompt or include them in scripts (see [“Using Remote CLI Commands in Scripts”](#) on page 25).

To execute a Remote CLI command from a Linux command prompt

- 1 Open a command prompt.
- 2 Execute the command including the connection parameters.

```
<command> <conn_params> <params>
```

Specify connection parameters in a configuration or pass them on the command line. The extension `.pl` is not required on Linux.

For example:

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

See “[Connection Options](#)” on page 22 for a complete list of connection parameters.

Uninstalling the Remote CLI Package

To uninstall the Remote CLI on a Linux system

- 1 Connect to the directory where you installed the Remote CLI.
- 2 Execute the `vmware-uninstall-rcli.pl` script.

The command uninstalls both the Remote CLI and the VI Perl Toolkit.

Installing and Executing Remote CLI Commands on Windows

When you install the Remote CLI package on Windows, the installation package includes all prerequisite software. The Remote CLI package is supported on the following Windows operating systems:

- Windows Vista SP 1
- Windows XP SP 2

Installing the Remote CLI Package

To install the Remote CLI Package on Windows

- 1 Download the Remote CLI Windows installer from <http://www.vmware.com/go/remotecli>.

- 2 Start the installer.

A warning message about the installer’s digital signature might appear.

- 3 Click **Yes** to ignore the warning message and continue with the installation.

If a previous version of the VI Perl Toolkit or the Remote CLI package exists on the target Windows system, the installer offers to repair or uninstall the previous version.

- 4 If prompted, accept the offer, or click **Next** in the Welcome page to continue.

- 5 If you do not want to install the Remote CLI in the default directory, click **Change** and choose a different directory.

The default location is C:\Program Files\VMware\VMware VI Remote CLI\bin.

- 6 Click **Next** to continue.
- 7 Click **Install** to proceed with the installation.

The installation might take several minutes to complete.

When the installation wizard finishes, you can test the installation by executing a Remote CLI command or by running one of the VI Perl Toolkit sample scripts or utility applications.

Executing Remote CLI Commands

After you install the Remote CLI, you can execute commands from the Windows command prompt or include them in scripts (see [“Using Remote CLI Commands in Scripts”](#) on page 25).

To execute a Remote CLI command from a Windows command prompt

- 1 Open a command prompt.
- 2 Navigate to the directory in which the Remote CLI is installed.
- 3 Execute the command, passing in the connection parameters and any other options.

```
cd C:\Program Files\VMware\VMware VI Remote CLI\bin
```

```
<command>.pl <conn_params> <params>
```

The extension .pl is required. For example:

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

See [“Connection Options”](#) on page 22 for a complete list of connection parameters.

Uninstalling the Remote CLI Package

You can uninstall the Remote CLI package like any other package.

To uninstall the Remote CLI on a Windows system

- 1 Choose **Start > Settings > Control Panel > Add or Remove Programs**.
- 2 In the panel that appears, choose VMwareVIRemoteCLI, and click **Remove**.
- 3 Click **Yes** when prompted.

The system uninstalls both the VI Perl Toolkit and the Remote CLI package.

Installing and Running the Remote CLI Virtual Appliance

A virtual appliance is a preconfigured virtual machine that packages applications with the operating system they require. When you power on the appliance, a Linux shell prompt allows you to run the commands on an ESX Server 3 host to which you connect remotely. The shell is a Linux bash shell in a pared-down CentOS Linux environment. All commands listed in this chapter are supported.

Installing and using the Remote CLI virtual appliance consists of these tasks:

- Task 1: “[Importing the Virtual Appliance](#)” on page 17
- Task 2: “[Running the Virtual Appliance](#)” on page 18

Importing the Virtual Appliance

You can import the virtual appliance from a file or directly from a URL or the VMware virtual appliance marketplace. To import from a file, download the virtual appliance before you start the import process from <http://www.vmware.com/go/remotecli>.

To import the virtual appliance

- 1 Using a VI Client, connect to a VirtualCenter Server or an ESX Server host.
- 2 In the inventory pane, select the import host for the appliance.
- 3 Choose **File > Virtual Appliance > Import**.

The Import Virtual Appliance wizard starts. You have these options:

- Click **Import from the VMware Virtual Appliance Marketplace** and click **Next**. Click **Remote CLI Appliance** and download the appliance.
 - Click **Import from File**, browse to an OVF file you already downloaded, and click **Next**.
 - Click **Import from URL**, supply the virtual appliance’s location and click **Next**.
- 4 Accept the license agreement.

- 5 Specify a name (optional), and choose a location for the virtual machine when prompted.

The wizard displays all data stores that are available and appropriate.

- 6 Choose the datastore to store the virtual machine on and click **Next**.

- 7 Review the information and click **Finish**.

The wizard creates an appliance virtual machine on the host you selected in Step 2. This might take a few minutes.

Running the Virtual Appliance

After the Import Virtual Machine wizard finishes, an appliance virtual machine appears in the VI Client inventory pane.

To run the virtual appliance

- 1 Select and power on the virtual machine.
- 2 Accept the End User License Agreement and provide a password for the root account to log in to the machine.
- 3 (Optional) If you want to give the virtual machine a static IP address, you can log in to the virtual machine console as the user network (same password as root) and set the IP address from there. By default, the virtual machine has a DHCP address.
- 4 After setting the IP address, run Remote CLI commands or Remote CLI scripts from the console.

You must supply connection information each time you run a command. See [“Connection Options”](#) on page 22.

Specifying Options for Remote CLI Commands

When you execute a Remote CLI command, you need to specify at a minimum the name of the execution server and the name of a user with login privileges and the corresponding password. You can provide the required parameters in several different ways.

At runtime, the appliance or Remote CLI package first processes any options that are set in the configuration file, next any environment variables, and finally command-line entries. This order of precedence always applies. That means, for example, that you cannot override an environment variable setting in a configuration file.

This section discusses how you can specify connection options using a session file, using environment variables, using a configuration file, or passing parameters at the command line. It also explains briefly how to use Windows Security Support Provider Interface (SSPI), which allows you to connect without specifying a user name or password.

Using a Session File

You can create a session file using the `save_session.pl` script included in the `apps/session` directory of the VI Perl Toolkit. You can then use that session file, which does not reveal password information, when you invoke the Remote CLI command. If the session file is not used for 30 minutes, it expires.

The VI Perl Toolkit is installed when you install a Remote CLI package and is also included in the Remote CLI appliance.

To create and use a session file

- 1 Connect to the directory where the script is located. For example, if running on Windows:

```
cd C:\Program Files\VMware\VMware VI Remote CLI\Perl\apps\session
```

- 2 Call `save_session.pl` which has the following syntax.

```
save_session.pl --savesessionfile <location> --server <vc_server>  
--vihost <esx_host>
```

You must supply connection parameters and the name of a session file in which the script saves an authentication cookie. For example:

```
save_session.pl --savesessionfile C:\Temp\my_session --server my_vc_server  
--vihost myserver
```

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

- 3 Pass in the session file using the `--sessionfile` parameter when you execute Remote CLI commands, as follows:

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

For example:

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

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

Setting Environment Variables

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

```
set VI_SERVER=<your_server_name>
```

NOTE Do not use escape characters in environment variables.

See [“Example: Adding a NAS Datastore to Multiple ESX Server Hosts”](#) on page 26.

Using a Configuration File

You use a text file that contains variable names and settings as a configuration file. Variables corresponding to the parameters shown in [Table 1-2, “Remote CLI Connection Options,”](#) on page 22.



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

Pass in the configuration file when you execute Remote CLI commands, as follows:

```
<command> --config <my_saved_config> --list
```

For example:

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

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

Use `--config` to save the configuration information in a different file than `./visdkrc`. If you specify `--config`, the system ignores the `./visdkrc` settings.

Passing Parameters at the Command Line

You can pass parameters at the command line using option name and option value pairs (some options have no value).

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

The following syntax results:

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

NOTE 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 (\) before special characters.

Using Microsoft Windows Security Support Provider Interface (SSPI)

You can use the `--passthroughauth` command-line argument to log in to a VirtualCenter Server (VirtualCenter version 2.5 Update 2 or later). Using `--passthroughauth` passes the credentials of the executing user to the server. If the executing user is known by both the machine from which you access the VirtualCenter Server and the machine running the VirtualCenter Server, no additional authentication is required.

If Remote CLI commands and the VirtualCenter Server run on the same machine, a local account for the executing user works. If they run on different machines, then the executing user must be an account in a domain trusted by both machines.

SSPI supports a number of protocols. By default, it chooses the `Negotiate` protocol, which indicates that client and server attempt to resolve a mutually supported protocol. Alternatively, you can use `--passthroughauthpackage` to specify another protocol supported by SSPI. Kerberos, the Windows standard for domain-level authentication, is commonly chosen. If the VirtualCenter Server is configured to accept only a specific protocol, specifying the protocol to Remote CLI commands with `--passthroughauthpackage` might be required for successful authentication to the server.

If you use `--passthroughauth`, you don't need to specify authentication information in any other way. You use the following at the command line:

```
<command> <login_params> --passthroughauth
```

See the Microsoft Web site for a detailed discussion of SSPI.

Available Options for Remote CLI Execution

[Table 1-2](#) lists options that are available for all Remote CLI commands. You use the parameter on the command line and the variable in configuration files.

Connection Options

[Table 1-2](#) lists connection options in alphabetical order. See “[Specifying Options for Remote CLI Commands](#)” on page 19 for the different approaches for specifying the parameters. [Table 1-3](#) lists a small number of common parameters that do not specify login information.

Table 1-2. Remote CLI Connection Options

Parameter and Environment Variable	Description
<code>--config <config_file></code> <code>VI_CONFIG=<config_file></code>	Uses the configuration file at the specified location. Specify a path that is readable from the current directory.
<code>--encoding <encoding></code> <code>VI_ENCODING=<encoding></code>	Specify the encoding to be used. One of <code>cp936</code> (Simplified Chinese) <code>ISO-8859-1</code> (German), or <code>Shift_JIS</code> (Japanese). You can use <code>--encoding</code> to specify the encoding the Remote CLI should map to when it is run on a foreign language system.
<code>--passthroughauth</code> <code>VI_PASSTHROUGHAUTH</code>	If you specify this option, the system uses the Microsoft Windows Security Support Provider Interface (SSPI) for authentication. You are not prompted for a user name and password. See the Microsoft Web site for a detailed discussion of SSPI. This option is supported only if you are running the Remote CLI on a Windows system and connecting to a VirtualCenter Server.
<code>--passthroughauthpackage <package></code> <code>VI_PASSTHROUGHAUTHPACKAGE=<package></code>	Use this parameter with <code>--passthroughauth</code> to specify a domain-level authentication protocol to be used by Windows. By default, SSPI uses the <code>Negotiate</code> protocol, which means that client and server attempt to negotiate a protocol that both support. If the VirtualCenter Server to which you are connecting is configured to use a specific protocol, you can specify that protocol using this parameter. This option is supported only if you are running the Remote CLI on a Windows system and connecting to a VirtualCenter Server.

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

Parameter and Environment Variable	Description
<code>--password <passwd></code> <code>VI_PASSWORD=<passwd></code>	<p>Use the specified password (used with <code>--username</code>) to log in to the server.</p> <ul style="list-style-type: none"> ■ If <code>--server</code> specifies a VirtualCenter Server, the user name and password apply to that server. You need no authentication to execute commands on the ESX Server hosts that server manages. ■ If <code>--server</code> specifies an ESX Server host, the user name and password apply to that server. <p>Use the empty string (' ' on Linux and " " on Windows) to indicate no password.</p> <p>If you do not specify a user name and password on the command line, the system prompts you and does not echo your input to the screen.</p>
<code>--portnumber <number></code> <code>VI_PORTNUMBER=<number></code>	Use the specified port to connect to the ESX Server host. Default is 443.
<code>--protocol <HTTP HTTPS></code> <code>VI_PROTOCOL=<HTTP HTTPS></code>	Use the specified protocol to connect to the ESX Server host. Default is HTTPS.
<code>--savesessionfile <file></code> <code>VI_SAVESESSIONFILE=<file></code>	Save a session to the specified file. The session expires if it has been unused for thirty minutes.
<code>--server <server></code> <code>VI_SERVER=<server></code>	<p>Use the specified ESX Server host or VirtualCenter Server host. Default is localhost.</p> <p>If <code>--server</code> points to a VirtualCenter Server, you use the <code>--vihost</code> option to specify the ESX Server host on which you want to execute the command. A command is supported for VirtualCenter if the <code>--vihost</code> option is defined.</p>
<code>--servicepath <path></code> <code>VI_SERVICEPATH=<path></code>	Use the specified service path to connect to the ESX Server host. Default is <code>/sdk/webService</code> .
<code>--sessionfile <file></code> <code>VI_SESSIONFILE=<file></code>	Use the specified session file to load a previously saved session. The session must be unexpired.
<code>--url <url></code> <code>VI_URL=<url></code>	Connect to the specified VI SDK URL.
<code>--username <u_name></code> <code>VI_USERNAME=<u_name></code>	<p>Use the specified user name.</p> <ul style="list-style-type: none"> ■ If <code>--server</code> specifies a VirtualCenter Server, the user name and password apply to that server. You need no authentication to execute commands on the ESX Server hosts that server manages. ■ If <code>--server</code> specifies an ESX Server host, the user name and password apply to that server. <p>If you do not specify a user name and password on the command line, the system prompts you and does not echo your input to the screen.</p>

Other Common Options

[Table 1-3](#) lists a small number of parameters you can use when invoking a Remote CLI command that are not used as login parameters.

Table 1-3. Remote CLI Common Parameters

Parameter	Description
<code>--help</code>	Print a brief usage message. The message lists first each command-specific option and then each of the common options.
<code>--verbose</code>	Display additional debugging information.
<code>--version</code>	Display version information.

Examples for Specifying Connection Options

The following examples illustrate using a session file, using environment variables in a `.visdkrc` file, executing a Remote CLI command on the command line, and using SSPI.

Example for Using a Session File

The following example first saves a session file, and then calls `vicfg-mpath` with its `--list` option. This example is for a Linux environment. See [“To create and use a session file”](#) on page 19 for a Windows example.

```
perl cd /usr/share/doc/vmware-viperl/samples/session
perl save_session.pl --savesessionfile /tmp/vimsession --server
    <my_vc_server> --username <username> --password <password>
    --vihost <my_server>

vicfg-mpath --sessionfile /tmp/vimsession --list
```

You can use the `save_session.pl` and `load_session.pl` scripts as examples for use of session files from scripts.

Example for Using a Configuration File

The following example shows the contents of a configuration file:

```
VI_SERVER = 10.17.211.138
VI_USERNAME = root
VI_PASSWORD = <root_password>
VI_PROTOCOL = https
VI_PORTNUMBER = 443
```

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

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

Example for Using the Command Line

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

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

Linux:

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

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

Example for Using an SSPI Protocol

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

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

See [“Using Microsoft Windows Security Support Provider Interface \(SSPI\)”](#) on page 21.

Using Remote CLI Commands in Scripts

Most administrators run scripts to perform the same task repeatedly or to perform a task on multiple hosts. This section presents a few typical scenarios and the corresponding scripts.

Example: Editing Files on the ESX Server Host

You cannot edit a file on the ESX Server 3i host directly because you have no service console access. You must retrieve the file from the host, make the change or changes, and then place the file on the ESX Server 3i host.

For example, to change the host agent log level, you must edit the `hostAgentConfig.xml` file. The following script uses `vifs` to download the host agent configuration file, uses `sed` to replace the log level with a user-supplied string, and replaces the file on the ESX Server host with the changed configuration file. The script then deletes temporary files it generated as part of the process.

```
HOST=your.hostname.com
```

```
vifs --server $HOST --username admin --password xxyyzz --get
    /host/hostAgentConfig.xml /tmp/ha.xml
sed -e "s#<level>.*</level>#<level>$1</level>#" < /tmp/ha.xml > /tmp/ha_new.xml
vifs --server $HOST --username admin --password xxyyzz --put /tmp/ha_new.xml
    /host/hostAgentConfig.xml
rm /tmp/ha.xml
rm /tmp/ha_new.xml
```

Example: Adding a NAS Datastore to Multiple ESX Server Hosts

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

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

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

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

```
#!/bin/sh
for i in {"esxi_server_a","esxi_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
```

Remote CLI Reference

2

This chapter is a reference to most of the Remote CLI commands you can use for administering your ESX Server hosts. Some commands with many options are discussed in separate chapters ([Chapter 3, “Managing Storage with vmkfstools,”](#) on page 63 and [Chapter 4, “Managing Virtual Machines with vmware-cmd,”](#) on page 77).

This chapter includes the following topics:

- [“Storage Management Commands”](#) on page 28
- [“Networking Commands”](#) on page 36
- [“System Management Commands”](#) on page 45
- [“Virtual Machine Management Commands”](#) on page 51
- [“Miscellaneous Management Commands”](#) on page 54
- [“Performing File System Operations with vifs”](#) on page 58
- [“Commands with an esxcfg Prefix”](#) on page 61

Storage Management Commands

[Table 2-1](#) lists the Remote CLI storage management commands discussed in this section.

Table 2-1. Storage Management Commands

Command	Description	See
<code>vicfg-dumppart</code>	Manages diagnostic partitions.	“Managing Diagnostic Partitions with vicfg-dumppart” on page 28.
<code>vicfg-mpath</code>	Configures storage arrays.	“Configuring Storage Arrays with vicfg-mpath” on page 30.
<code>vicfg-nas</code>	Manages NAS file systems.	“Managing NAS File Systems with vicfg-nas” on page 33.
<code>vicfg-rescan</code>	Rescans the storage configuration.	“Rescanning with vicfg-rescan” on page 34.
<code>vicfg-vmhbadevs</code>	Finds available LUNs on the ESX Server host.	“Finding Available LUNs with vicfg-vmhbadevs” on page 35.

Managing Diagnostic Partitions with `vicfg-dumppart`

The `vicfg-dumppart` command can query, set, and scan an ESX Server host’s diagnostic partitions. See *“Creating a Diagnostic Partition”* in the *ESX Server 3i Configuration Guide* and the *ESX Server 3 Configuration Guide*.

When you run `vicfg-dumppart`, the partition’s console name is not displayed. Because the chosen partition is activated, the active and configured states are redundant and are not displayed.

Options for `vicfg-dumppart`

[Table 2-2](#) lists the command-specific options for `vicfg-dumppart`. You must also specify connection options. See *“Remote CLI Connection Options”* on page 22.

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

Option	Description
<code>--activate <partition></code> <code>-a <partition></code>	Activates the configured diagnostic partition specified by <code><partition></code> . This option has the same effect as <code>--set</code> .
<code>--deactivate [<partition>]</code> <code>-d [<partition>]</code>	Deactivates the currently active diagnostic partition. The option also unsets the dump partition. Specifying the name of the partition is optional. WARNING: If you run <code>vicfg-dumppart</code> with this option, your system cannot write errors to a file until another partition is activated. You lose any error record if errors occur.
<code>--find</code> <code>-f</code>	Finds all diagnostic partitions on the ESX Server host and prints them in this order: Parallel adapter, block adapter, Fibre Channel, hardware iSCSI, software iSCSI. The partitions well suited as diagnostic partitions are listed before the less suitable partitions.
<code>--get-active</code> <code>-t</code>	Displays the active diagnostic partition for this system. Running <code>vicfg-dumppart</code> with this option returns the internal name of the partition (<code>vmhbaX:X:X:X</code>) or <code>none</code> if no partition is set.
<code>--list</code> <code>-l</code>	Lists all partitions on the system that have the appropriate partition type to act as an ESX Server diagnostic partition. CAUTION: Execution might take several minutes and slow down your ESX Server host because the command scans all LUNs on the system.
<code>--set vmhba<X:X:X:X></code> <code>-s vmhba<X:X:X:X></code>	Sets the active and configured diagnostic partition for this system using the <code>vmhba</code> name of the partition to use. When you set the diagnostic partition, it is activated. There is no distinction between an active and a configured diagnostic partition. The <code>--activate</code> option is included only for backward compatibility.
<code>--vihost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vihost</code> to specify the ESX Server host to execute the command on.

Examples for `vicfg-dumppart`

The following examples for `vicfg-dumppart` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

Shows the current diagnostic partition the VMkernel uses.

```
vicfg-dumppart.pl <conn_params> -s vmhba1:0:0:6
```

Sets the active partition to `vmhba1:0:0:6`. After the command has been executed, `-c` reports `vmhba1:0:0:6` as the active partition.

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

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

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

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

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

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

```
Partition number 5 on vml.01000000005550543650334130314844334d4150333336
-> vmhba1:0:0:5 -> /dev/sda5 Partition number 6 on
vml.01000000005550543650334130314844334d4150333336 -> vmhba1:0:0:6 ->
/dev/sda6 Partition number 7 on
vml.01000000005550543650334130314844334d4150333336 -> vmhba1:0:0:7 ->
/dev/sda7 Partition number 1 on
vml.010000000033485a394757375a535433373333 -> vmhba1:3:0:1 -> /dev/sdb1
```

Configuring Storage Arrays with `vicfg-mpath`

The `vicfg-mpath` command can configure multipath settings for Fibre Channel or iSCSI LUNs. For more information on multipathing, see “Managing Multiple Paths” in the *ESX Server 3 Configuration Guide* and the *ESX Server 3i Configuration Guide*. The *Fibre Channel SAN Configuration Guide* and the *iSCSI SAN Configuration Guide* discuss some storage-specific aspects of multipathing.

Names of virtual machine HBAs are not guaranteed to be valid across reboots. Use VML LUN names to be sure of consistency. The VML name of a LUN is the unique name given to that LUN by VMware. This name is globally unique to a LUN and remains associated with a LUN across reboots.

Options for `vicfg-mpath`

[Table 2-3](#) lists the command-specific options for `vicfg-mpath`.

NOTE If you are changing the preferred path or if you change 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.
-

You must also specify connection options. See [“Connection Options”](#) on page 22.

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

Option	Description
<code>--bulk</code> <code>-b</code>	Shows all LUNs and paths in a format that scripts can parse easily.
<code>--detailed</code> <code>-d</code>	Shows all information about a LUN and its paths including the LUN's VML name. A LUN's VML name is a unique name VMware assigns to the LUN. This name is globally unique to a LUN and remains associated with a LUN across reboots.
<code>--hbas</code> <code>-a</code>	Prints the list of HBAs that are identifiable by a unique ID. This includes Fibre Channel and iSCSI devices. Parallel and block devices do not appear in this list.
<code>--list</code> <code>-l</code>	Lists all LUNs on the system and the paths to these LUNs through adapters. For each LUN, this command displays the type, internal name, console name, size, paths, and path selection policy.
<code>--lun=<lun></code> <code>-L=<lun></code>	Required to specify the LUN to use in operations. This option is a required parameter for other options and is not used by itself.
<code>--path=<path></code> <code>-P=<path></code>	Required to specify the path to use in operations. This option is a required parameter for other options and is not used by itself. The path name can be either the long path UID or the shorter runtime name of the path. You can use <code>--path</code> to filter what the list commands return.
<code>--policy [mru fixed]</code> <code>-p [mru fixed]</code>	Set the policy for a given LUN to <code>mru</code> or <code>fixed</code> . This option requires the <code>--lun</code> option. <ul style="list-style-type: none"> ■ Most Recently Used (<code>mru</code>) selects the most recently used path to send I/O to a device. ■ Fixed (<code>fixed</code>) uses only the active path. Two additional policies, round robin (<code>rr</code>) and custom are available on an experimental basis. See the technical note <i>Round Robin Load Balancing</i> .
<code>--preferred</code> <code>-f</code>	Set the specified path to be the preferred path for a specified LUN. When you set this option, you must also set the <code>--lun</code> and <code>--path</code> options.
<code>--query</code> <code>-q</code>	Query a specific LUN for its information and print the information. When you set this option, you must also set the <code>--lun</code> option.
<code>--state [on off]</code> <code>-s [on off]</code>	Set the state of a given LUN path to either <code>on</code> or <code>off</code> . This option requires that both the <code>--lun</code> and <code>--path</code> options are also set.
<code>--vihost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vihost</code> to specify the ESX Server 3 host to execute the command on.

Examples for vicfg-mpath

The following examples for `vicfg-mpath` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vicfg-mpath.pl <conn_params> -l
```

Displays all available paths.

```
vicfg-mpath.pl <conn_params> -q --lun=vml.123456
```

Displays the paths for disk `vml.123456`.

```
vicfg-mpath.pl <conn_params> --policy=mru --lun=vmhba0:0:1
```

Sets the path policy for disk `vmhba0:0:1` to `mru`.

```
vicfg-mpath.pl <conn_params> --path=vmhba1:0:1 --lun=vmhba0:0:1 --state=on
```

Enables a path for disk `vmhba0:0:1`.

```
vicfg-mpath.pl <conn_params> --path=vmhba0:1:1 --state=off --lun=vmhba0:0:1  
-p fixed
```

Disables a path and sets the policy to `fixed` for disk `vmhba0:0:1`.

```
vicfg-mpath.pl <conn_params> -l
```

Lists all LUNs on the system and the paths to these LUNs through adapters.

```
vicfg-mpath.pl <conn_params> -a
```

Displays LUN information. Results in output like the following.

```
vmhba2 2305843973628581845 42:2.0
vmhba3 2305843973628747050 4c:00.0
vmhba4 2306125448607554858 4c:00.1
vmhba5 50:1.1
```

```
vicfg-mpath.pl <conn_params> -b
```

Returns a bulk path list suitable for parsing.

```
vicfg-mpath.pl <conn_params> --policy fixed --path vmhba2:0:1 --lun vmhba2:0:1  
--preferred
```

Sets the preferred path. Results in output like the following:

```
Setting vmhba2:0:1 -- vmhba2:0:1 as preferred path
Setting vmhba2:0:1 policy to fixed
```

Managing NAS File Systems with vicfg-nas

The `vicfg-nas` command can manipulate NAS file systems associated with your ESX Server host. For more information on working with NAS file systems, see “Network Attached Storage” in the *ESX Server 3 Configuration Guide* and the *ESX Server 3i Configuration Guide*.

Options for vicfg-nas

[Table 2-4](#) lists the command-specific options for `vicfg-nas`. You must also specify connection options. See “[Remote CLI Connection Options](#)” on page 22.

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

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

Examples for vicfg-nas

The following examples for `vicfg-nas` require connection options. See “[Specifying Options for Remote CLI Commands](#)” on page 19.

```
vicfg-nas.pl <conn_params> -l
```

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

```
vicfg-nas.pl <conn_params> -a
```

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

```
vicfg-nas.pl <conn_params> -d
```

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

```
vicfg-nas.pl <conn_params> -a -s new_share
```

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

Rescanning with vicfg-rescan

After certain storage management operations, you must perform a rescan operation using `vicfg-rescan`. See “Performing a Rescan” in the *ESX Server 3i Configuration Guide* and the *ESX Server 3 Configuration Guide*. The *Fibre Channel SAN Configuration Guide* discusses rescan on Fibre Channel storage in detail.

When you perform a rescan on an ESX Server host, the command returns only an indication of success or failure and no detailed information.

Options for vicfg-rescan

[Table 2-5](#) lists the command-specific options for `vicfg-rescan`. You must also specify connection options. See “[Remote CLI Connection Options](#)” on page 22.

Table 2-5. Options for `vicfg-rescan`

Option	Description
<vmkernel_SCSI_adapter_name>	Name of the adapter to scan, for example, <code>vmhba0</code> .
<code>--vihost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vihost</code> to specify the ESX Server host to execute the command on.

Example for vicfg-rescan

The following example for `vicfg-rescan` requires connection options. See “[Connection Options](#)” on page 22.

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

Results in output like the following:

```
Scan operation succeeded
```

Finding Available LUNs with vicfg-vmhbadevs

The `vicfg-vmhbadevs` command can display information about available LUNs on ESX Server hosts.

Options for vicfg-vmhbadevs

Table 2-6 lists the command-specific options for `vicfg-nas`. You must also specify connection options. See “Remote CLI Connection Options” on page 22.

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

Option	Description
<code>--query</code> <code>-q</code>	Prints the output in 2.5 compatibility mode.
<code>--vihost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vihost</code> to specify the ESX Server host to execute the command on.
<code>--vmfs</code> <code>-m</code>	Prints the VMFS UUID as well as the <code>vmhba</code> and <code>/dev</code> names if the LUN is a VMFS volume.

Examples for vicfg-vmhbadevs

The following examples for `vicfg-vmhbadevs` require connection options. See “Specifying Options for Remote CLI Commands” on page 19. The examples illustrate the output that might result from executing `vicfg-vmhbadevs`.

```
#vicfg-vmhbadevs.pl <conn_params> -q
vmhba0:1:1
/vmfs/devices/disks/vml.0100000005550543650334330323656424d4150333336
```

```
#vicfg-vmhbadevs.pl <conn_params> --vmfs
/vmfs/devices/disks/vml.0100000005550543650334330323656424d4150333336
46f14706-2083a0f5-491f-001b7803ba96
```

```
#vicfg-vmhbadevs.pl <conn_params> -m vmhba1:0:0:2
vmhba1:0:0:2
/vmfs/devices/disks/vml.0100000005550543650334330323656424d4150333336
46f14706-2083a0f5-491f-001b7803ba96
```

Networking Commands

Table 2-7 lists Remote CLI networking commands discussed in this section. For an introduction to networking in the Virtual Infrastructure, see Chapter 2, “Networking,” in the *ESX Server 3i Configuration Guide* and the *ESX Server 3 Configuration Guide*.

Table 2-7. Remote CLI Networking Commands

Command	Description	See
vicfg-dns	Specifies an ESX Server host’s DNS configuration.	“ Specifying the DNS Configuration with vicfg-dns ” on page 36
vicfg-nics	Manages the ESX Server host’s physical NICs.	“ Managing Physical NICs with vicfg-nics ” on page 38
vicfg-ntp	Specifies the NTP server.	“ Specifying the NTP Server with vicfg-ntp ” on page 39
vicfg-route	Manipulates the ESX Server host’s route entry.	“ Manipulating the route Entry with vicfg-route ” on page 40
vicfg-vmknic	Adds, deletes, and modifies VMkernel NICs.	“ Managing VMkernel NICs with vicfg-vmknic ” on page 41
vicfg-vswitch	Adds or removes virtual switches or modifies virtual switch settings.	“ Managing Virtual Switches with vicfg-vswitch ” on page 42

Specifying the DNS Configuration with vicfg-dns

The `vicfg-dns` command can list and specify your ESX Server host’s DNS configuration. Call the command without any command-specific options to list the existing DNS configuration.

Options for vicfg-dns

Table 2-8 lists the command-specific options for `vicfg-dns`. You must also specify connection options. See “[Remote CLI Connection Options](#)” on page 22.

Table 2-8. Options for vicfg-dns

Option	Description
<code>--hostname <h_name></code> <code>-n <h_name></code>	Human-readable host name portion of the DNS name.
<code>--domain <d_name></code> <code>-d <d_name></code>	Domain name portion for this ESX Server host.
<code>--dhcp yes no</code> <code>-H yes no</code>	Indicates whether or not the ESX Server host uses DHCP (dynamic host control protocol) to automatically determine DNS configuration. You must specify the string <code>yes</code> or the string <code>no</code> .

Table 2-8. Options for `vicfg-dns` (Continued)

Option	Description
<code>--dns <server_list></code> <code>-D <server_list></code>	DNS server or servers to be used. Specify a comma-separated list of DNS servers, in order of preference.
<code>--vihost</code> <code>-h</code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vihost</code> to specify the ESX Server host to execute the command on.
<code>--vmnic</code> <code>-V</code>	The virtual network adapter to use when overriding the system DNS. This parameter is required when <code>--dhcp</code> is <code>yes</code> . For ESX Server hosts, it must be one of the Service Console network adapters and for ESX Server 3i hosts, it must be one of the VMkernel network adapter. The specified virtual network adapter must have DHCP configured.

Examples for `vicfg-dns`

The following examples for `vicfg-dns` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

Configures the host name for the specified server.

```
vicfg-dns.pl <conn_params> -d mydomain.biz
```

Configures the domain name for the specified server.

```
vicfg-dns.pl <conn_params> -H yes
```

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

```
vicfg-dns.pl <conn_params>
```

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 Server host. For example:

```
Host Name      localhost
Domain Name   localdomain
DHCP          true
DNS Servers   10.17.XXX.1
              10.17.XXX.2
```

Managing Physical NICs with vicfg-nics

The `vicfg-nics` command can manage physical NICs, that is, the Ethernet switches used by an ESX Server host. The `--list` option prints the VMkernel name for the network adapter, its PCI ID, driver, link state, speed, duplex, and a short PCI description of the card.

You can also use `vicfg-nics` to specify speed and duplex settings for a NIC.

Options for vicfg-nics

[Table 2-9](#) lists the command-specific options for `vicfg-nics`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

Table 2-9. Options for `vicfg-nics`

Option	Description
<code>--auto</code> <code>-a</code>	Sets the specified network adapter to auto-negotiate its speed and duplex settings.
<code>--duplex [full half] <nic></code> <code>-d [full half] <nic></code>	Sets the duplex value at which a given network adapter should run to either <code>full</code> (transmit data in both directions at the same time) or <code>half</code> (transmit data in one direction at a time).
<code>--list</code> <code>-l</code>	Lists the network adapters in the system, and prints their current and configured speed and duplex information.
<code>--speed <speed> <nic></code> <code>-s <speed> <nic></code>	Sets the speed at which a given network adapter should run. Valid values for <code><speed></code> are 10, 100, 1000, or 10000.
<code>--vhost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vhost</code> to specify the ESX Server host to execute the command on.

Examples for vicfg-nics

The following examples for `vicfg-nics` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

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

```
vicfg-nics.pl <conn_params> -a vmknic2
```

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

```
vicfg-nics.pl <conn_params> -l
```

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

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

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

Specifying the NTP Server with vicfg-ntp

The `vicfg-ntp` command allows you to specify the NTP (Network Time Protocol) server for an ESX Server host. Some protocols, such as Kerberos, must have accurate information about the current time.

To set up NTP for your ESX Server

- 1 Call `vicfg-ntp --add` to add the NTP server.
- 2 Call `vicfg-ntp --start` to start the service or `vicfg-ntp --list` to list the service.

Options for vicfg-ntp

[Table 2-10](#) lists the command-specific options for `vicfg-ntp`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

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

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

Examples for vicfg-ntp

The following examples for `vicfg-ntp` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vicfg-ntp.pl <conn_params> -l
```

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

```
vicfg-ntp.pl <conn_params> --start
```

Starts the NTP service on the specified host.

```
vicfg-ntp.pl <conn_params> -a 192.168.XXX.XXX
```

Adds the host specified by 192.168.XXX.XXX as a remote NTP server.

Manipulating the route Entry with vicfg-route

The `vicfg-route` command can list or set the default IP gateway. Changing the gateway might be required if you move your ESX Server host to a new physical location. The command supports a subset of the Linux `route` command’s options.

If no options are specified, the command prints the default gateway. You can set the default gateway by executing `vicfg-route <gateway>`.

Options for vicfg-route

[Table 2-11](#) lists the command-specific options for `vicfg-route`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

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

Option	Description
<code><gateway></code>	The IP address or the host name of the machine that should be set as the gateway for the VMkernel IP stack.
<code>--vhost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vhost</code> to specify the ESX Server host to execute the command on.

Examples for vicfg-route

The following examples for `vicfg-route` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vicfg-route.pl <conn_params>
```

Lists the VMkernel IP stack’s default gateway entry.

vicfg-route.pl <conn_params> 10.17.XXX.XXX

Sets the default gateway for the ESX Server host specified in <conn_params> to 10.17.XXX.XXX.

Managing VMkernel NICs with vicfg-vmknic

The `vicfg-vmknic` command can configure virtual network adapters.

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

Options for vicfg-vmknic

[Table 2-12](#) lists the command-specific options for `vicfg-vmknic`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

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

Option	Description
<port-group>	Required argument. Specifies the port group. The name is the same as the VMkernel NIC name.
<code>--add</code> <code>-a</code>	Adds a virtual network adapter to the system. You must specify IP parameters and a port group name. The newly added virtual network adapter is enabled when the command completes successfully.
<code>--delete <port_group></code> <code>-d <port_group></code>	Deletes the virtual network adapter on the specified port group.
<code>--ip <ipaddress> DHCP</code> <code>-i <ipaddress> DHCP</code>	Sets the IP address (X.X.X.X) to be used for the virtual network adapter. When you set an IP address, you must specify the <code>--netmask</code> option in the same command. If you specify DHCP instead of an IP address, DHCP must be supported in the VMkernel.
<code>--list</code> <code>-l</code>	Lists virtual network adapters on the system. The list contains the network information, port group, MTU, and current state for each virtual network adapter in the system.
<code>--netmask <netmask></code> <code>-n</code>	IP netmask (X.X.X.X) to be used for the virtual network adapter. When you set a netmask, you must specify the <code>--ip</code> option in the same command.
<code>--vhost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vhost</code> to specify the ESX Server host on which you want to execute the command.

Examples for vicfg-vmknic

The following examples for `vicfg-vmknic` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vicfg-vmknic.pl <conn_params> -a -ip 192.168.XXX.XXX
-n 255.255.255.0 "VMkernel NIC Name"
```

Adds the specified VMkernel NIC to the system.

```
vicfg-vmknic.pl <conn_params> -d "VMkernel NIC Name"
```

Deletes the specified VMkernel NIC from the system.

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

Modifies the VMkernel NIC's IP address and netmask.

```
vicfg-vmknic.pl <conn_params> --ip DHCP "VMkernel NIC Name"
```

Sets the specified VMkernel NIC to use DHCP.

Managing Virtual Switches with vicfg-vswitch

The `vicfg-vswitch` command can add or remove virtual switches or modify virtual switch settings. A virtual switch is an abstracted network device. It can route traffic internally between virtual machines and link to external networks. See *“Virtual Switches”* in the *ESX Server 3i Configuration Guide* and the *ESX Server 3 Configuration Guide*.

By default, each virtual machine has a single virtual switch called `vSwitch0`.

Options for vicfg-vswitch

[Table 2-13](#) lists the command-specific options for `vicfg-vswitch`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

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

Option	Description
<code>--add <vswitch_name></code> <code>-a <vswitch_name></code>	Adds the specified virtual switch to the system.
<code>--add-pg <portgroup> <switch></code> <code>-A <portgroup> <switch></code>	Adds a port group to the specified virtual switch.

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

Option	Description
<code>--check <virtual_switch></code> <code>-c <virtual_switch></code>	Check whether a virtual switch exists. Prints 1 if the switch exists and prints 0 otherwise. Use the virtual switch name, e.g. <code>vSwitch0</code> or <code>vSwitch1</code> , to specify the virtual switch.
<code>--check-pg <port_group></code> <code>-C <port_group></code>	Checks whether the specified port group exists.
<code>--delete <vswitch_name></code> <code>-d <vswitch_name></code>	Deletes a virtual switch. Executing the command with this option fails if any ports on the virtual switch are still in use by VMkernel networks, <code>vswifs</code> , or virtual machines.
<code>--del-pg <portgroup></code> <code>-D <portgroup></code>	Deletes a port group from the virtual switch. Executing the command with this option fails if the port group is in use, for example, by a virtual machine or a VMkernel network.
<code>--link <pnic></code> <code>-L <pnic></code>	Adds an uplink adapter (physical NIC) to a virtual switch. Executing the command with this option attaches a new unused physical network adapter to a virtual switch.
<code>--list</code> <code>-l</code>	Lists all virtual switches and their port groups.
<code>--mtu <virtual_switch></code> <code>-m <virtual_switch></code>	Sets the MTU (maximum transmission unit) of the virtual switch. This option affects all physical NICs assigned to the virtual switch.
<code>--pg <port group></code> <code>-p <port group></code>	Provides the name of the port group for the <code>--vlan</code> option. Specify <code>ALL</code> to set VLAN IDs on all port groups of a virtual switch.
<code>--unlink <pnic></code> <code>-U <pnic></code>	Removes a physical NIC from a virtual switch. A physical NIC is a physical Ethernet adapter to which the virtual switch is connected. If you remove the last NIC, you lose physical network connectivity for that switch.
<code>--vihost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vihost</code> to specify the ESX Server host to execute the command on.
<code>--vlan</code> <code>-v</code>	Sets the VLAN ID for a specific port group of a virtual switch. Setting the option to 0 disables the VLAN for this port group. If you specify this option, you must also specify the <code>--pg</code> option.

Examples for vicfg-vswitch

The following examples for `vicfg-vswitch` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vicfg-vswitch.pl <conn_params> --add vSwitch1
```

Adds vSwitch1 as a virtual switch.

```
vicfg-vswitch.pl <conn_params> --add-pg=group1 vSwitch0
```

Adds a port group to vSwitch0.

```
vicfg-vswitch.pl <conn_params> -c vSwitch0
```

Checks whether vSwitch0 exists. Prints 1 if the switch exists, 0 if the switch does not exist.

```
vicfg-vswitch.pl <conn_params> -m 9000 vSwitch0
```

Sets the MTU of the virtual switch vSwitch0 to 9000.

```
vicfg-vswitch.pl <conn_params> -l
```

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

Sample Output from vicfg-vswitch -l

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

Switch Name	Num Ports	Used Ports	MTU	Uplinks
vSwitch0	64	5	1500	vmnic2,vmnic0

PortGroup Name	VLAN ID	Used Ports	Uplinks
group1	0	0	vmnic0,vmnic2
group2	0	0	vmnic0,vmnic2
group3	0	1	vmnic0,vmnic2

Switch Name	Num Ports	Used Ports	MTU	Uplinks
vSwitch1	64	1	1500	

PortGroup Name	VLAN ID	Used Ports	Uplinks
bldg1	0	0	
bldg2	0	0	
bldg3	0	1	

System Management Commands

A number of Remote CLI commands allow you to perform system configuration and performance monitoring.

Table 2-14. System Management Commands

Command	Description	See
resxstop	Monitors in real time how ESX Server hosts use resources.	“Using resxstop for Performance Monitoring” on page 45
vicfg-cfgbackup	Backs up the configuration data of an ESX Server 3i system and restores previously saved configuration data.	“Using vicfg-cfgbackup for Configuration Information” on page 45
vicfg-snmp	Sets up SNMP (Simple Network Management Protocol).	“Setting Up SNMP with vicfg-snmp” on page 47
vicfg-syslog	Specifies the syslog server and the port to connect to that server for ESX Server 3i hosts.	“Specifying the syslog Server with vicfg-syslog” on page 49
vihostupdate	Performs maintenance on ESX Server 3i hosts.	“Performing Maintenance with vihostupdate” on page 50

Using resxstop for Performance Monitoring

The `resxstop` Remote CLI command allows you to examine how ESX Server hosts use resources. You can use the utility in one of three modes: interactive (default), batch, or replay. The *Resource Management Guide* explains how to use `resxstop` in each of these modes and provides information about available commands and display statistics.

Using vicfg-cfgbackup for Configuration Information

The `vicfg-cfgbackup` command can back up ESX Server 3i configuration data and restore them later.

VMware recommends that you back up your host configuration after you change the configuration or upgrade the ESX Server 3i image.

The *ESX Server 3i Embedded Setup Guide* and *ESX Server 3i Installable Setup Guide* discuss backing up and restoring ESX Server 3i configuration.

NOTE This command is not available for ESX Server 3 version 3.5 and cannot be executed from the VirtualCenter Server.

Options for vicfg-cfgbackup

Table 2-15 lists the command-specific options for `vicfg-cfgbackup`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

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

Option	Description
<code>--force</code> <code>-f</code>	Forces the restore of the configuration.
<code>--load <backupfile></code> <code>-l <backupfile></code>	Restores configuration from <code><backupfile></code> onto the host.
<code>--quiet</code> <code>-q</code>	Does not prompt for user confirmation.
<code>--reset</code> <code>-r</code>	Resets the host, that is, restores to factory settings.
<code>--save <backupfile></code> <code>-s <backupfile></code>	Backs up the host configuration to <code><backupfile></code> . 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 the Remote CLI from the Remote CLI virtual appliance, the backup file is saved locally on the virtual appliance. Local storage for backup files is safe because virtual appliances are stored in the directory <code>/vmfs/volumes/<datastore></code> , which is separate from the ESX Server 3i image and configuration files.

Examples for vicfg-cfgbackup

The following examples for `vicfg-cfgbackup` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

Backs up the host configuration.

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

Restores the host to factory settings.

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

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

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

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

Setting Up SNMP with vicfg-snmp

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

ESX Server 3i ships with an SNMP management agent different from the agent that runs in the ESX Server 3 service console. Currently, the ESX Server 3i SNMP agent supports only SNMP traps, not GETS and is off by default.

This command is not supported on VirtualCenter Server.

To set up the SNMP agent

- 1 Enable the SNMP service (`vicfg-snmp --enable`).
- 2 Specify at least one community (`vicfg-snmp --communities`).
- 3 Configure a trap destination (`vicfg-snmp --targets`).

Options for vicfg-snmp

[Table 2-16](#) lists the command-specific options for `vicfg-snmp`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

Table 2-16. vicfg-snmp Command Options

Option	Description
<code>--communities <comm1>[,...]</code> <code>-c <comm1>[,...]</code>	Specifies communities separated by commas. The settings specified using this option overwrite any previous settings.
<code>--disable</code> <code>-D</code>	Stops the SNMP service.
<code>--enable</code> <code>-E</code>	Starts the SNMP service.
<code>--port <port-number></code> <code>-p <port-number></code>	Sets the port used by the SNMP agent. The default is UDP 162.
<code>--reset</code> <code>-r</code>	Clears all previously-specified communities and targets.
<code>--show</code> <code>-s</code>	Displays the current SNMP configuration.

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

Option	Description
<code>--targets <hostname[@port]> </community>[,...]</code> <code>-t <hostname[@port]> </community>[,...]</code>	Set the destination for (notifications) traps. The settings specified using this flag overwrite any previous settings.
<code>--test</code> <code>-T</code>	Send a test notification that can be used to validate the SNMP configuration.

Examples for vicfg-snmp

The following examples for `vicfg-snmp` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

Sets the community to public.

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

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

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

Enables the SNMP service.

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

Sets the SNMP agent port to port 163.

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

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

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

Specifying the syslog Server with vicfg-syslog

The `vicfg-syslog` command allows you to specify a remote syslog server for an ESX Server 3i host. You cannot execute this command on an ESX Server version 3.5 host or a VirtualCenter Server.

The *Basic System Administration* manual discusses system logs in more detail and explains how to set them up using the VI Client.

Options for vicfg-syslog

[Table 2-17](#) lists the command-specific options for `vicfg-syslog`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

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

Option	Description
<code>--setport -p</code>	Sets the port for the syslog server. Can be used with <code>-s</code> .
<code>--setserver <sys_server></code> <code>-s <sys_server></code>	Host name or IP address of the remote syslog server. Can be used with <code>-p</code> .
<code>--show</code> <code>-i</code>	Displays the remote syslog server, if a syslog server has been set up.
<code>--vihost <host></code> <code>-h <host></code>	When you execute a Remote CLI with the <code>--server</code> option pointing to a VirtualCenter Server host, use <code>--vihost</code> to specify the ESX Server host to execute the command on.

Examples for vicfg-syslog

The following examples for `vicfg-syslog` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

Displays the syslog server configuration.

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

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

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

Sets the port number used by the syslog server.

Performing Maintenance with vihostupdate

The `vihostupdate` command allows you to perform maintenance of your ESX Server 3i hosts. The command can install software updates, enforce software update policies, and track installed software.

NOTE The `vihostupdate` command is supported only on ESX Server 3i. For maintenance of your ESX Server 3 version 3.5 host, use the `esxupdate` command. This command is not supported on VirtualCenter Server.

Software updates might be patches for addressing critical security issues or urgent bug fixes, or they might be general updates or maintenance releases. They might be located on the local file system or on an NFS, FTP, or HTTP server. Each update consists of a descriptor file and a set of packages. The descriptor controls the installation process and checks that requirements are met. For example, you might be required to power off all virtual machines running on the server you are about to update, or you might need to reboot the server after the update.

For a detailed discussion of `vihostupdate`, see the *ESX Server 3i Setup Guide*.

To update the firmware of an ESX Server 3i host

- 1 Download the update bundle to the machine where you run the `vihostupdate` command.
- 2 Call the `vihostupdate` command.

The `vihostupdate` options allow you to either work with the bundle directly, or point to a metadata file.

The update process first pushes the update bundle to the host, and then requests that the host perform the update.

Options for vihostupdate

[Table 2-18](#) lists the command-specific options for `vihostupdate`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

Table 2-18. Options for `vihostupdate`

Option	Description
<code>--bundle <bundle_file_name></code> <code>-b <bundle_file_name></code>	Unpack the downloaded bundle ZIP file. If you specify this option, you cannot specify <code>--metadata</code> .
<code>--install</code> <code>-i</code>	Patch the host with applicable packages in the update bundle. This option takes no arguments, but you must also include either <code>-b</code> to specify a bundle, or <code>-m</code> to specify a metadata file.

Table 2-18. Options for `vihostupdate` (Continued)

Option	Description
<code>--metadata <metadata_xml_file></code> <code>-m <metadata_xml_file></code>	Path to the <code>metadata.xml</code> file that contains information about the update bundle. If you specify this option, you cannot specify <code>--bundle</code> .
<code>--query</code> <code>-q</code>	List packages installed on the host. This option returns version information for the ESX Server host, as well as all packages installed and their version numbers.
<code>--scan <dir></code> <code>-s <dir></code>	Scan for packages that apply to the host in the directory specified by <code><dir></code> .

Examples for `vihostupdate`

The following examples for `vihostupdate` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

Assume the path to an update bundle is `EESX-142-test-release.zip`. If you then go to the directory where you copied that bundle, you can execute the following commands:

```
vihostupdate.pl <conn_params> -i -b EESX-142-test-release.zip
```

Unpacks and patches the host

```
vihostupdate.pl <conn_params> -b EESX-142-test-release.zip
```

Unpacks the file, but does not patch the host.

```
vihostupdate.pl <conn_params> -i -m EESX-142-test-release.zip/metadata.xml
```

Patches a host using the file `EESX-142-test-release.zip`.

Running the first example is equivalent to running the second and third in sequence.

Virtual Machine Management Commands

A number of Remote CLI commands allow you to perform virtual machine management.

Table 2-19. Virtual Machine Management Commands

Command	Description	See
<code>svmotion</code>	Moves a virtual machine’s configuration file and optionally its disks while the virtual machine is running.	“Migrating Virtual Machines Using <code>svmotion</code>” on page 52
<code>vmware-cmd</code>	Performs virtual machine operations remotely.	Chapter 4, “Managing Virtual Machines with <code>vmware-cmd</code>,” on page 77.

Migrating Virtual Machines Using `svmotion`

The `svmotion` command moves a virtual machine's configuration file, and, optionally, its disks, while the virtual machine is running. Storage VMotion™ is discussed in some detail in *Basic System Administration*. You can use `svmotion` to initiate migrations on both ESX Server 3i and ESX Server 3 hosts. You always call this command with the `--server` connection option pointing to a VirtualCenter Server.

You can specify the virtual machine and its destination with the `--vm` option. By default, all virtual disks are relocated to the same datastore as the virtual machine. You can use the `--disks` option to relocate individual virtual disks to different datastores.

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

You can run the `svmotion` command either in interactive or noninteractive mode.

- To use the command in interactive mode, type `svmotion --interactive`. You are prompted for all the information necessary to complete the storage migration. When you invoke the command in interactive mode, all other parameters are ignored.
- In noninteractive mode, the `svmotion` command uses the following syntax:

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

Square brackets indicate optional elements.

On Windows systems, use double quotes instead of single quotes around the virtual machine and disks parameter.

Options for `svmotion`

[Table 2-20](#) lists the command-specific options for `svmotion`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

Table 2-20. Options for `svmotion`

Option	Description
<datacenter>	The datacenter that contains the virtual machine to be migrated. You must quote the name if it contains white spaces or other special characters.
<VM config datastore path>	Datastore path to the virtual machine's configuration file. If the path contains spaces or other special characters, you must quote it. See “To determine the path to the virtual machine's configuration file” on page 53.

Table 2-20. Options for `svmotion` (Continued)

Option	Description
<code><new datastore></code>	Name of the new datastore to which the virtual machine configuration file or disk is to be moved. Do not include brackets around the name of the datastore.
<code>--disks</code>	If you do not specify this parameter, all virtual disks associated with a virtual machine are relocated to the same datastore as the virtual machine configuration file. By specifying this parameter, you can locate individual virtual disks to different datastores. If you would like to keep a virtual disk on its current datastore, use the <code>--disks</code> option for that disk with its current datastore as the <code><new datastore></code> .
<code><virtual disk datastore path></code>	Used with <code>--disks</code> to specify the location of a virtual disk file for which you want to override the default move (see the syntax listed above this table). See “To determine the path to a virtual disk file” on page 53.

Determining Parameter Information

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

To determine the path to the virtual machine’s configuration file

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

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

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

To determine the path to a virtual disk file

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

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

Examples for svmotion

The following examples for `svmotion` require connection options. See [“Connection Options”](#) on page 22. Each example should use one line, even though line breaks are included for legibility below. Square brackets in the examples indicate a datastore.

```
svmotion.pl <conn_params> --interactive
```

Starts the interactive version of the `svmotion` command.

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

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

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

Relocates a virtual machine's configuration file to `new_datastore`, but leave the two disks (`myvm_1.vmdk` and `myvm_2.vmdk`) in `old_datastore`.

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

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

Miscellaneous Management Commands

The Remote CLI includes the following miscellaneous commands, discussed in this section.

Table 2-21. Miscellaneous Management Commands

Command	Description	See
<code>vicfg-advcfg</code>	Performs advanced configuration. Use this command as instructed by VMware.	“Using vicfg-advcfg in Special Circumstances” on page 55
<code>vicfg-module</code>	Enables VMkernel options. Use this command with the options listed in this document, or instructed by VMware.	“Enabling VMkernel Module Options with vicfg-module” on page 55
<code>vicfg-user</code>	Creates, modifies, deletes, and lists local direct access users and groups of users.	“Managing Users and Groups with vicfg-user” on page 56

Using vicfg-advcfg in Special Circumstances

The `vicfg-advcfg` command performs a number of low-level advanced options and is 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.

Enabling VMkernel Module Options with vicfg-module

The `vicfg-module` command allows you to set and get VMkernel options. The command is usually used when VMware Technical Support, a Knowledge Base article, or VMware documentation instruct you to do so.

This command is not supported on VirtualCenter Server.

Options for vicfg-module

[Table 2-22](#) lists the command-specific options for `vicfg-module`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

Table 2-22. Options for `vicfg-module`

Option	Description
<code>--set-options <module_name> -s</code>	Specifies the option string to be passed to the module when the module is loaded.
<code>--get-options <module_name> -g <module_name></code>	Returns the option string configured to be passed to the module when the module is loaded. This string is not necessarily the option string currently in use by the module.

Examples for vicfg-module

The following examples illustrate NetQueue configuration. The *ESX Server 3i Configuration Guide* and the *ESX Server 3 Configuration Guide* explains NetQueue configuration in more detail. The following examples require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

Configures a supported NIC to use NetQueue.

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

Verifies that the NetQueue module has been configured.

Managing Users and Groups with vicfg-user

An ESX Server grants access to its resources when a known user with appropriate permissions logs on to the host with a password that matches the one stored for that user. The `vicfg-user` command allows you to create, modify, delete, and list local direct access users and groups of users on an ESX Server host. You cannot execute this command on a VirtualCenter Server.

User management is discussed in detail in the *ESX Server 3 Configuration Guide*, the *ESX Server 3i Configuration Guide*, and the *Basic System Administration* document.

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

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

Options for vicfg-user

[Table 2-23](#) lists the command-specific options for `vicfg-user`. You must also specify connection options. See [“Remote CLI Connection Options”](#) on page 22.

Table 2-23. Options for `vicfg-user`

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

Table 2-23. Options for `vicfg-user` (Continued)

Option	Description
<code>--operation</code> <code>-o</code>	Operation to perform. Specify <i>add</i> , <i>modify</i> , <i>delete</i> , or <i>list</i> .
<code>--removegroup <group_list></code> <code>-G <group_list></code>	Comma-separated list of groups to remove the target user from.
<code>--removeuser <user_list></code> <code>-U <user_list></code>	Comma-separated list of users to be removed from the target group.
<code>--role</code> <code><administrator read-only no-access></code> <code>-r</code> <code><administrator read-only no-access></code>	Role for the target user or group. Specify one of <i>administrator</i> , <i>read-only</i> , or <i>no-access</i> .
<code>--shell</code> <code>-s</code>	Grant shell access to the target user. Valid values are <i>yes</i> and <i>no</i> . Default is no shell access. Use this command to change the default, or to revoke shell access rights after they have been granted. This option is supported only on ESX Server 3. The option is not supported on ESX Server 3i.

Examples for `vicfg-user`

The following examples for `vicfg-user` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

Adds with login ID `user27`.

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

Modifies the user with login ID `user27`.

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

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

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

Removes the user with login ID `user27`.

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

Adds `group42` as a group.

```
vicfg-user.pl <conn_params> -e group -o add -d group42 -u test
```

Adds a user `test` to the group `group42`.

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

Removes the group group42.

Performing File System Operations with vifs

The `vifs` command performs common operations such as copy, remove, get, and put on files and directories. The command is supported on ESX Server 3 but not on VirtualCenter Server.

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

```
http://ESX3ihost_IP_Address/host
http://ESX3ihost_IP_Address/folder
```

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

File and Directory Groups

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

- Host – Host configuration files. You must specify the file’s unique name identifiers. You specify host locations using the `host/<path>` syntax.
- Temp – The `/tmp` directory and files inside that directory. You 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) or `/folder/testvms/VM1/VM1.vmx&dsName=myStorage1`
(Windows)

The two example paths refer to the same virtual machine configuration file for the virtual machine VM1 inside the myStorage1 datastore’s testvms/VM1 directory.

NOTE Because directory names often use special characters or spaces, enclosing the path in quotes is highly recommended for both styles.

Running vifs

When you run `vifs`, you can specify the operation name and argument and also one of the standard connection options discussed in [Table 1-2](#). Use aliases, symbolic links, or wrapper scripts to simplify the invocation syntax.

NOTE The `vifs` command has no concept of a working directory or last directory or file operated on.

Options for vifs

[Table 2-24](#) lists the command-specific options for `vifs`. All `vifs` options work on datastore files or directories. Some options also work on host files and files in the `temp` directory. You must also specify connection options. See “[Remote CLI Connection Options](#)” on page 22.

Table 2-24. Options for `vifs`

Command	Description	For...	Examples
<code>--copy</code> <code>-c <source></code> <code><target></code>	Copies a file in a datastore to another location in a datastore. The <code><source></code> must be a remote source path, the <code><target></code> a remote target path or directory. The <code>--force</code> option replaces existing destination files.	Datastore Temp	<code>copy src_file_path</code> <code>dst_directory_path</code> <code>[--force]</code> <code>copy src_file_path</code> <code>dst_file_path [--force]</code>
<code>--dir</code> <code>-D</code> <code><remote_dir></code>	Lists the contents of a datastore directory.	Datastore Temp	<code>dir</code> <code>datastore_directory_path</code>
<code>--force</code> <code>-F</code>	Overwrites the destination file. Used with <code>--move</code> .	Datastore Temp	<code>copy src_file_path</code> <code>dst_file_path [--force]</code>
<code>--get</code> <code>-g</code> <code><remote_path></code> <code><local_path></code>	Downloads a file from the ESX Server 3 host to the machine on which you run the Remote CLI. This operation uses HTTP GET.	Datastore Host	<code>get src_dstore_file_path</code> <code>dst_local_file_path</code> <code>get src_d store_dir_path</code> <code>dst_local_file_path</code>
<code>--listdc</code> <code>-C</code>	Lists the datacenter paths available on a server.	Datastore Host	
<code>--listds</code> <code>-S</code>	Lists the datastore names on the server. When multiple data centers are available, you can use the <code>--dc</code> (<code>-Z</code>) argument to specify the name of the datacenter from which you want to list the datastore.	Datastore Host	<code>vifs --listds</code>

Table 2-24. Options for `vifs` (Continued)

Command	Description	For...	Examples
<code>--mkdir</code> <code>-M</code> <code><remote_dir></code>	Creates a directory in a datastore. This operation fails if the parent directory of <code>dst_datastore_file_path</code> does not exist.	Datastore Temp	<code>mkdir dst_directory_path</code>
<code>--move</code> <code>-m</code> <code><source></code> <code><target></code>	Moves a file in a datastore to another location in a datastore. The <code><source></code> must be a remote source path, the <code><target></code> a remote target path or directory. The <code>--force</code> option replaces existing destination files.	Datastore Temp	<code>move src_file_path dst_directory_path [--force]</code> <code>move src_file_path dst_file_path [--force]</code>
<code>--put</code> <code>-p</code> <code><local_path></code> <code><remote_path></code>	Uploads a file from the machine on which you run the Remote CLI to the ESX Server 3i host. This operation uses HTTP PUT. This command can replace existing host files but cannot create new files.	Datastore Host, Temp	<code>put src_local_file_path dst_file_path</code> <code>put src_local_file_path dst_directory_path</code>
<code>--rm</code> <code>-r</code> <code><remote_path></code>	Deletes a datastore file.	Datastore Temp	<code>rm dst_file_path</code>
<code>--rmdir</code> <code>-R</code> <code><remote_dir></code>	Deletes a datastore directory. This operation fails if the directory is not empty.	Datastore Temp	<code>rmdir dst_directory_path</code>

Examples for `vifs`

This section gives some examples for `vifs`. Executing `vifs` works only when you are connected directly to an ESX Server 3i host. The command does not work when you connect to a VirtualCenter Server and attempt to connect to the ESX Server 3i host through the VirtualCenter Server.

You can execute the commands listed below on a Linux system. For corresponding commands on a Windows system, use double quotes instead of single quotes.

The following examples for `vifs` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

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

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

Lists the contents of directory `dir_3`.

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

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

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

Creates the directory called `new_dir`.

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

Places a copy of the local `test_put` file into the specified server's `new_dir` directory.

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

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

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

Removes the `new_dir` folder.

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

Retrieves the file `my_text` from the ESX Server 3i host and places it in the local machine's `root` folder.

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

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

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

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

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

Commands with an `esxcfg` Prefix

For several of the commands listed in this chapter, you might have used corresponding service console commands starting with an `esxcfg` prefix in scripts to manage ESX Server 3.0. To facilitate easy migration from ESX Server 3.0 to ESX Server 3i or ESX Server 3 version 3.5, the commands with the `esxcfg` prefix are available as Remote CLI commands.

NOTE VMware recommends that you use the command with the `vicfg` prefix. Commands with the `esxcfg` prefix are available mainly for compatibility reasons and might become obsolete.

Table 2-25 lists all Remote CLI commands for which a command with an `esxcfg` prefix is available.

Table 2-25. Commands with an `esxcfg` Prefix

Command with <code>vicfg</code> prefix	Command with <code>esxcfg</code> prefix	See
<code>vicfg-advcfg</code>	<code>esxcfg-advcfg</code>	“Using <code>vicfg-advcfg</code> in Special Circumstances” on page 55.
<code>vicfg-cfgbackup</code>	<code>esxcfg-cfgbackup</code>	“Using <code>vicfg-cfgbackup</code> for Configuration Information” on page 45
<code>vicfg-dns</code>	<code>esxcfg-dns</code>	“Specifying the DNS Configuration with <code>vicfg-dns</code>” on page 36
<code>vicfg-dumppart</code>	<code>esxcfg-dumppart</code>	“Managing Diagnostic Partitions with <code>vicfg-dumppart</code>” on page 28.
<code>vicfg-module</code>	<code>esxcfg-module</code>	See “Enabling VMkernel Module Options with <code>vicfg-module</code>” on page 55
<code>vicfg-mpath</code>	<code>esxcfg-mpath</code>	“Configuring Storage Arrays with <code>vicfg-mpath</code>” on page 30.
<code>vicfg-nas</code>	<code>esxcfg-nas</code>	“Managing NAS File Systems with <code>vicfg-nas</code>” on page 33.
<code>vicfg-nics</code>	<code>esxcfg-nics</code>	“Managing Physical NICs with <code>vicfg-nics</code>” on page 38.
<code>vicfg-ntp</code>	<code>esxcfg-ntp</code>	“Specifying the NTP Server with <code>vicfg-ntp</code>” on page 39
<code>vicfg-rescan</code>	<code>esxcfg-rescan</code>	“Rescanning with <code>vicfg-rescan</code>” on page 34.
<code>vicfg-route</code>	<code>esxcfg-route</code>	“Manipulating the route Entry with <code>vicfg-route</code>” on page 40.
<code>vicfg-snmp</code>	<code>esxcfg-snmp</code>	“Setting Up SNMP with <code>vicfg-snmp</code>” on page 47
<code>vicfg-syslog</code>	<code>esxcfg-syslog</code>	“Specifying the syslog Server with <code>vicfg-syslog</code>” on page 49
<code>vicfg-user</code>	<code>esxcfg-user</code>	“Managing Users and Groups with <code>vicfg-user</code>” on page 56
<code>vicfg-vmhbadevs</code>	<code>esxcfg-vmhbadevs</code>	“Finding Available LUNs with <code>vicfg-vmhbadevs</code>” on page 35.
<code>vicfg-vmknic</code>	<code>esxcfg-vmknic</code>	“Managing VMkernel NICs with <code>vicfg-vmknic</code>” on page 41.
<code>vicfg-vswitch</code>	<code>esxcfg-vswitch</code>	“Managing Virtual Switches with <code>vicfg-vswitch</code>” on page 42.

Managing Storage with vmkfstools

3

You use the `vmkfstools` Remote CLI to create and manipulate virtual disks, file systems, logical volumes, and physical storage devices on an ESX Server host. Using `vmkfstools`, you can create and manage a virtual machine file system (VMFS) on a physical partition of a disk. You can also use `vmkfstools` to manipulate files, such as virtual disks, stored on VMFS-3 and NFS.

NOTE The `vmkfstools` command on ESX Server does not support all the options supported in the ESX Server 3 version 3.5 service console.

You can execute the `vmkfstools` Remote CLI like other Remote CLI commands. Specify the ESX Server host on which to execute the command, and specify additional connection options as discussed in [“Specifying Options for Remote CLI Commands”](#) on page 19. This command is not supported on VirtualCenter Server.

In most cases, you don’t need to log in as the root user to run the `vmkfstools` command. However, some commands, such as the file system commands, might require the root user login.

You can perform most `vmkfstools` operations using the VI Client instead of the command line. For information on using the VI Client to work with storage, see *“Configuring Storage”* in the *ESX Server 3 Configuration Guide* and the *ESX Server 3i Configuration Guide*.

This chapter includes the following topics:

- [“vmkfstools Command Syntax”](#) on page 64
- [“vmkfstools File System Options”](#) on page 67
- [“vmkfstools Virtual Disk Options”](#) on page 69

vmkfstools Command Syntax

The `vmkfstools` command has the following syntax:

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

Here, `<options>` can be one or more command-line options and associated values (see [“Specifying Options”](#) on page 64) and `<target>` is the partition, device, or path to which you want to apply the command (see [“Specifying the Target”](#) on page 66).

Specifying Options

You can use one or more command-line options and associated arguments to specify what you want to do. For example, when you create a new virtual disk, you can specify the disk format.

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

```
vmkfstools --createfs vmfs3 --blocksize 2m vmhba1:3:0:1
vmkfstools -C vmfs3 -b 2m vmhba1:3:0:1
```

[Table 3-1](#) gives an overview of `vmkfstools` file system options. It includes a brief description and a pointer to where the option is discussed.

Table 3-1. vmkfstools File System Option Overview

Option	Description	See
<code>--blocksize</code> <code>-b</code>	Uses the specified size for file system creation. Used with <code>--createfs</code> .	“Creating a VMFS File System” on page 67
<code>--createfs</code> <code>-C</code>	Creates a VMFS file system.	“Creating a VMFS File System” on page 67.
<code>--extendfs</code> <code>-Z</code>	Extends existing VMFS-3 file system.	“Extending an Existing VMFS-3 Volume” on page 68.
<code>--human-readable</code> <code>-j</code>	Lists attributes in human-readable format.	“Listing Attributes of a VMFS Volume” on page 69
<code>--queryfs</code> <code>-P</code>	Lists attributes of file system.	“Listing Attributes of a VMFS Volume” on page 69.
<code>--setfsname</code> <code>-S</code>	Sets the label for the file system. Used with <code>--createfs</code> .	“Creating a VMFS File System” on page 67

[Table 3-1](#) gives an overview of `vmkfstools` file system options. It includes a brief description and a pointer to where the option is discussed.

Table 3-2. vmkfstools Virtual Disk Option Overview

Option	Description	See
--adaptype -a	Uses the specified type for disk creation. Used with -c and -i.	“Creating a Virtual Disk” on page 70.
--clonevirtualdisk -i	Clones the specified virtual disk.	“Cloning a Virtual or Raw Disk” on page 73.
--createdrm -r	Maps a raw disk to a file on a VMFS file system.	“Creating a Virtual Compatibility Mode Raw Device Mapping” on page 75.
--createrdmpassthru -z	Maps a passthrough raw disk to a file on a VMFS file system.	“Creating a Physical Compatibility Mode Raw Device Mapping” on page 76.
--createvirtualdisk -c	Creates a virtual disk.	“Creating a Virtual Disk” on page 70.
--deletevirtualdisk -U	Deletes the specified virtual disk.	“Deleting a Virtual Disk” on page 72.
--diskformat -d	Uses the specified format for disk creation. Used with -c and -i.	“Supported Disk Formats” on page 69.
--extendvirtualdisk -X	Extends the specified virtual disk.	“Extending a Virtual Disk” on page 74.
--geometry -g	Displays virtual disk geometry.	“Displaying Virtual Disk Geometry” on page 76.
--inflatedisk -j	Converts a thin virtual disk to eagerzeroedthick format, preserving all existing data.	“Inflating a Thin Virtual Disk” on page 72.
--migratevirtualdisk -M	Migrates a VMFS-2 disk to VMFS-3.	“Creating a Virtual Compatibility Mode Raw Device Mapping” on page 75.
--queryrdm -q	Lists the attributes of a raw disk mapping.	“Listing Attributes of an RDM” on page 76.
--renamevirtualdisk -E	Renames the specified virtual disk.	“Renaming a Virtual Disk” on page 73.
--writezero -w	Cleans the virtual disk by writing zeros over all its data.	“Initializing a Virtual Disk” on page 71.

Specifying the Target

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

File System Target

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

- To specify a VMFS file system, use this format:

```
/vmfs/volumes/<file_system_UUID>
```

or

```
/vmfs/volumes/<file_system_label>
```

- To specify a VMFS file, use this format:

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

You don't need to enter the entire path if the current working directory is the parent directory of myDisk.vmdk.

For example,

```
/vmfs/volumes/datstore1/rh9.vmdk
```

See [“vmkfstools File System Options”](#) on page 67.

Disk Partition Target

Specify a disk partition using a vmhbaA:T:L:P format, where A, T, L, and P are integers representing adapter, target, LUN, and partition number respectively. The partition digit must be greater than zero and should correspond to a valid VMFS partition of type fb. For example, vmhba0:2:3:1 refers to the first partition on LUN 3, target 2, HBA 0.

See [“vmkfstools Virtual Disk Options”](#) on page 69.

Device Target

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

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

vmkfstools File System Options

This section discusses `vmkfstools` file system options. Some options are for advanced users only. You must also specify connection options. See “[Remote CLI Connection Options](#)” on page 22.

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

```
vmkfstools.pl <conn_params> --createfs vmfs3 --blocksize 2m vmhba1:3:0:1
vmkfstools.pl <conn_params> -C vmfs3 -b 2m vmhba1:3:0:1
```

File system options allow you to create a VMFS file system. These options do not apply to NFS file systems. You can perform many of these tasks through the VI Client.

Creating a VMFS File System

The `-C` option creates a VMFS-3 file system on the specified SCSI partition, such as `vmhba1:0:0:1`. The partition becomes the file system's head partition.

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

NOTE You can have only one VMFS volume for a LUN.

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

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

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

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

Examples for Creating a VMFS File System

The following examples require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vmkfstools.pl <conn_params> -C vmfs3 -b 1m -S
my_vmfs/vmfs/devices/disks/vmhba1:3:0:1
```

Creates a new VMFS-3 file system named `my_vmfs` on the first partition of target 3, LUN 0 of SCSI adapter 1. The file block size is 1MB.

```
vmkfstools.pl <conn_params> -C vmfs3 -S my_vmfs vmhba1:0:0:4
```

```
vmkfstools.pl <conn_params> --createfs vmfs3 --setfsname my_vmfs vmhba1:0:0:4
```

```
vmkfstools.pl <conn_params> --createfs vmfs3 --blocksize 1m--setfsname my_vmfs
vmhba1:0:0:4
```

```
vmkfstools.pl <conn_params> --createfs vmfs3 -b 4m --setfsname my_vmfs
vmhba1:0:0:4
```

Extending an Existing VMFS-3 Volume

The `-Z` option adds an extent to a previously created VMFS volume `<existing-VMFS-volume>`.

```
-Z --extendfs <extension-device> <existing-VMFS-volume>
```

Each time you use this option, you extend a VMFS-3 volume with a new extent so that the volume spans multiple partitions. A logical VMFS-3 volume can have at most 32 physical extents.



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

Example for Extending an Existing Volume

The following example requires connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vmkfstools.pl <conn_params> -Z /vmfs/devices/disks/vmhba0:1:2:1
/vmfs/devices/disks/vmhba0:3:0:1
```

Extends the logical file system by allowing it to span to a new partition. The extended file system spans two partitions, `vmhba1:3:0:1` and `vmhba0:1:2:1`. In this example, `vmhba1:3:0:1` is the name of the head partition.

Listing Attributes of a VMFS Volume

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

```
-P --queryfs
```

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

Example for Listing Attributes

The following example requires connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

This command might return the following:

```
VMFS-3.31 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:
    vmhba2:0:0:6
```

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

vmkfstools Virtual Disk Options

Virtual disk options allow you to set up, migrate, and manage virtual disks stored in VMFS-2, VMFS-3, and NFS file systems. You can also perform most of these tasks through the VI Client.

Supported Disk Formats

When you create or clone a virtual disk, you can use the `-d --diskformat` suboption to specify the format for your disk. Choose from the following formats:

- `zeroedthick` (default) – Space required for the virtual disk is allocated during creation. Any data remaining on the physical device is not erased during creation, but will be zeroed out on demand at a later time on first write from the virtual machine. The virtual machine does not read stale data from disk.

- **eagerzeroedthick** – Space required for the virtual disk is allocated at creation time. In contrast to **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.
- **thick** – Space required for the virtual disk is allocated during creation. This type of formatting doesn't zero out any old data that might be present on this allocated space. Only the root user is allowed to create disks in this format.
- **thin** – Thin-provisioned virtual disk. Unlike with the **thick** format, space required for the virtual disk is not allocated during creation, but is supplied, zeroed out, on demand at a later time.
- **rdm** – Virtual compatibility mode raw disk mapping.
- **rdmp** – Physical compatibility mode (pass-through) raw disk mapping.
- **raw** – Raw device.
- **2gbsparse** – A sparse disk with 2GB maximum extent size. You can use disks in this format with other VMware products. However, you cannot power on a sparse disk on an ESX Server host unless you first reimport the disk in a compatible format, such as **thick** or **thin**, with **vmkfstools**.
- **monospase** – A monolithic sparse disk. You can use disks in this format with other VMware products.
- **monoflat** – A monolithic flat disk. You can use disks in this format with other VMware products.

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

With **-i --clonevirtualdisk**, **vmkfstools** accepts **rdm:dev**, **rdmp:dev**, **raw:dev**, **thin**, **2gbsparse**, **monoflat**, and **monospase**.

With NFS files, **vmkfstools** supports only **thin**, **thick**, **zeroedthick**, and **2gbsparse**. Because the NFS server and not the ESX Server host decides the allocation policy, **thick**, **zeroedthick**, and **thin** usually have the same result. The default allocation policy on most NFS servers is **thin**.

Creating a Virtual Disk

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

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

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

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

- `-a` specifies the device driver that is used to communicate with the virtual disks. You can choose between BusLogic and LSI Logic SCSI drivers.
- `-d` specifies disk formats. For detailed description of the disk formats, see [“Supported Disk Formats”](#) on page 69.

Examples for Creating a Virtual Disk

The following examples require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vmkfstools.pl <conn_params> -c 2048m /vmfs/volumes/my_vmfs/myOS.vmdk
```

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

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

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

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

Creates a virtual disk associated with the specified adapter.

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

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

Initializing a Virtual Disk

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

```
-w --writezeros
```

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



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

Examples for Initializing a Virtual Disk

The following examples require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

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

Inflating a Thin Virtual Disk

The `-j` option converts a thin virtual disk to eagerzeroedthick format, preserving all existing data.

```
-j --inflatedisk
```

See [“Supported Disk Formats”](#) on page 69.

Examples for Inflating a Virtual Disk

The following examples require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vmkfstools.pl <conn_params> --inflatedisk '[myVMFS] testsep1.vmdk'
```

```
vmkfstools.pl <conn_params> -j '[myVMFS] test02.vmdk'
```

```
vmkfstools.pl <conn_params> --inflatedisk -a buslogic
/vmfs/volumes/myvmfs/thin.vmdk
```

Deleting a Virtual Disk

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

```
-U --deletevirtualdisk
```

Example for Deleting a Virtual Disk

The following example requires connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

Deletes the virtual disk `test.vmdk`.

Renaming a Virtual Disk

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

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

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

Examples for Renaming a Virtual Disk

The following examples require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

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

```
vmkfstools.pl <conn_params> 10.20.120.196 --renamevirtualdisk
/vmfs/volumes/myvmfs/my_OS.vmdk /vmfs/volumes/myvmfs/my_new_OS.vmdk
```

Cloning a Virtual or Raw Disk

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

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

For ESX Server, you must specify the `--diskformat` and `--adapertype` options with `-i`. The `--diskformat` option specifies the disk format for the copy you create. See [“Supported Disk Formats”](#) on page 69.

NOTE To clone the ESX Server host’s Redo logs while preserving their hierarchy, use the `vimfs -C` command.

Example for Cloning a Virtual or Raw Disk

The following example requires connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

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

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

Migrating VMware Workstation and VMware GSX Server Virtual Machines

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

To migrate VMware Workstation and GSX Server virtual machines

- 1 Use `vmkfstools` to import a VMware Workstation or GSX Server disk into your `/vmfs/volumes/myVMFS/` directory or any subdirectory.
- 2 In the VI Client, create a new virtual machine using the **Custom** configuration option.
- 3 When configuring a disk, select **Use an existing virtual disk** and attach the VMware Workstation or GSX Server disk you imported.

Extending a Virtual Disk

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

```
-X --extendvirtualdisk <newSize>[k|m|g]
```

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

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

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

Examples for Extending a Virtual Disk

The following example requires connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vmkfstools.pl <conn_params> 10.20.120.132 -X 5g <disk name>.dsk
```

Extends a 4GB virtual disk by 1GB.

```
vmkfstools.pl <conn_params> 10.20.120.132 -X 50M
/vmfs/volumes/my_newVMFS/my_disk.vmdk
```

Creating a Virtual Compatibility Mode Raw Device Mapping

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

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

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

When specifying the `<device>` parameter, enter `0` for the partition to indicate that the entire raw disk is used. Use the following format:

```
/vmfs/devices/disks/vmhbaA:T:L:0
```

For details on configuring and using RDMs, see *“Raw Device Mapping”* in the *ESX Server 3 Configuration Guide* and the *ESX Server 3i Configuration Guide*.

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

Examples For Creating a Virtual Compatibility Mode RDM

The following examples require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vmkfstools.pl <conn_params> -r /vmfs/devices/disks/vmhba2:1:0:0
/vmfs/volumes/storage1/rdm210.vmdk
```

Creates a virtual compatibility mode RDM file `/vmfs/volumes/storage1/rdm210.vmdk` and maps the `/vmfs/devices/disks/vmhba2:1:0:0` raw disk to that file.

```
vmkfstools.pl <conn_params> -r /vmfs/devices/disks/vmhba1:3:0:0 my_rdm.vmdk
```

Creates an RDM file named `my_rdm.vmdk` and maps the `vmhba1:3:0:0` raw disk to that file. You can configure a virtual machine to use the `my_rdm.vmdk` mapping file by adding the following lines to the virtual machine configuration file:

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

Creating a Physical Compatibility Mode Raw Device Mapping

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

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

The mapping lets a virtual machine bypass ESX Server 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. When specifying the `<device>` parameter, enter `0` for the partition to indicate that the entire raw device is used. Use the following format:

```
/vmfs/devices/disks/vmhba:T:L:0
```

Example for Creating a Physical Compatibility Mode RDM

The following example requires connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vmkfstools.pl <conn_params> -z /vmfs/devices/disks/vmhba2:1:0:0
/vmfs/volumes/storage1/rdmpass.vmdk
```

Creates a physical compatibility mode RDM file named `rdmpass.vmdk` and maps the `vmhba2:1:0:0` to that file. You can't use the name of a file that already exists.

Listing Attributes of an RDM

The `-q` option lets you list the attributes of an RDM.

```
-q --queryrdm
```

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

Displaying Virtual Disk Geometry

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

```
-g --geometry
```

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

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

Managing Virtual Machines with `vmware-cmd`

4

This chapter discusses the `vmware-cmd` Remote CLI command, which allows you to manage the virtual machines in your VMware Infrastructure environment.

You can use `vmware-cmd` to perform various operations on a virtual machine, including registering a virtual machine on the local server, getting the power state of a virtual machine, setting configuration variables, and so on.

NOTE Older versions of `vmware-cmd` supported a set of connection options and general options that differ from the options in other Remote CLI commands. The `vmware-cmd` Remote CLI supports the existing service console-based `vmware-cmd` syntax and option names for backward compatibility. The Remote CLI also supports the standard Remote CLI `--server`, `--username`, and `--password` options, but none of the other connection options.

This chapter includes the following topics:

- [“Connection Options for `vmware-cmd`”](#) on page 78
- [“General Options for `vmware-cmd`”](#) on page 78
- [“Server Options for `vmware-cmd`”](#) on page 79
- [“Virtual Machine Options for `vmware-cmd`”](#) on page 79
- [“`vmware-cmd` Examples”](#) on page 83

Connection Options for vmware-cmd

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

Table 4-1. vmware-cmd Connection Options

Option	Description
<code>--server <host></code> <code>-H <host></code>	Target ESX Server host or VirtualCenter Server host.
<code>-T <target></code>	When you execute <code>vmware-cmd</code> with the <code>-H</code> option pointing to a VirtualCenter Server, use <code>-T</code> to specify the ESX Server host to execute the command on.
<code>-O <port></code>	Alternative connection port. The default port number is 902.
<code>--username <username></code> <code>-U <username></code>	User name for a user who is authorized to log in to the host specified by <code>-H</code> or <code>-T</code> .
<code>--password <password></code> <code>-P <password></code>	Password for the user specified by <code>-U</code> .
<code>-Q <protocol></code>	Protocol to use, either <code>http</code> or <code>https</code> . Default is <code>https</code> .

General Options for vmware-cmd

The `vmware-cmd` Remote CLI command supports the following general options.

Table 4-2. vmware-cmd General Options

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

Server Options for vmware-cmd

The following options allow you to perform operations on the ESX Server host you are connected to. When specifying `vm-path`, choose one of the following formats:

- Datastore prefix style: `'[ds_name] relative_path'`
for example `'[myStorage1] testvms/vm1/VM1.vmx'` (Linux) or
`"[myStorage1] testvms\vm1\VM1.vmx"` (Windows)
- UUID-based path: `folder/subfolder/file`
for example `'/vmfs/volumes/mystorage/Ftestvms/vm1/VM1.vmx'` (Linux) or
`"/vmfs/volumes/mystorage/testvms/vm1/VM1.vmx"` (Windows)

Table 4-3. vmware-cmd Server Operations

Server Operation	Description
<code>-l</code>	Lists the registered virtual machines on the server.
<code>-s register <vm-path></code> <code>[datacenter]</code> <code>[resource_pool]</code>	Registers a virtual machine specified by <code><vm-path></code> on the server. You can optionally specify a datacenter or resource pool.
<code>-s unregister <vm-path></code>	Unregisters a virtual machine specified by <code><vm-path></code> on the server.

Virtual Machine Options for vmware-cmd

The following options allow you to perform operations on virtual machines on the ESX Server you are connecting to.

When specifying `vm-path`, choose one of the following formats:

- Datastore prefix style: `'[ds_name] relative_path'`
for example `'[myStorage1] testvms/vm1/VM1.vmx'` (Linux) or
`"[myStorage1] testvms\vm1\VM1.vmx"` (Windows)
- UUID-based path: `folder/subfolder/file`
for example `'/vmfs/volumes/mystorage/Ftestvms/vm1/VM1.vmx'` (Linux) or
`"/vmfs/volumes/mystorage/testvms/vm1/VM1.vmx"` (Windows)

Table 4-4. Commands that apply to any VMware Infrastructure Host

Command	Description
<vm-path> answer	Prompts the user to answer a question for a virtual machine waiting for user input.
<vm-path> connectdevice <device_name>	Connects the specified virtual device to a virtual machine.
<vm-path> createsnapshot <name> <description> <quiesce> <memory>	<p>Creates a snapshot of the specified virtual machine, providing a name and description for the snapshot.</p> <ul style="list-style-type: none"> ■ If the <quiesce> flag is 1 and the virtual machine is powered on when the snapshot is taken, VMware Tools is used to quiesce the file system in the virtual machine. Quiescing a file system is a process of bringing the on-disk data of a physical or virtual computer into a state suitable for backups. This process might include such operations as flushing dirty buffers from the operating system's in-memory cache to disk, or other higher-level application-specific tasks. ■ If the <memory> flag is 1, then a dump of the internal state of the virtual machine is included in the snapshot. Memory snapshots take longer to create.
<vm-path> disconnectdevice <device_name>	Disconnects the specified virtual device from the virtual machine.
<vm-path> getconfigfile	Returns a string containing the name of the virtual machine's configuration file. This command fails if the virtual machine is not connected.
<vm-path> getguestinfo <variable>	Retrieves the value for a GuestInfo variable.
<vm-path> getproductinfo <prodinfo>	<p>Returns information about the product, where <prodinfo> is product, platform, build, majorversion (product's major version number), minorversion (product's minor version number) or revision.</p> <p>If product is specified, the return value is one of the following: ws (VMware Workstation), gsx (VMware GSX Server) esx (VMware ESX Server) or unknown (unknown product type).</p> <p>If platform is specified, the return value is one of the following: windows (Microsoft Windows), linux (Linux operating system) or unknown (unknown platform type).</p>
<vm-path> getstate	Retrieves the execution state of a virtual machine. The state can be on, off, suspended, stuck (requires user input), or unknown.

Table 4-4. Commands that apply to any VMware Infrastructure Host (Continued)

Command	Description
<vm-path> gettoolslastactive	Returns an integer indicating how much time has passed, in seconds, since the last heartbeat was detected from the VMware Tools service. This value is initialized to zero when the virtual machine powers on. It stays at zero until the first heartbeat is detected, after which the value is always greater than zero until the virtual machine is power cycled again.
<vm-path> getuptime	Accesses the uptime of the guest operating system on the virtual machine.
<vm-path> hassnapshot	Returns 1 if the virtual machine already has a snapshot. Returns 0 otherwise.
<vm-path> removesnapshots	Removes all snapshots belonging to the virtual machine. If no snapshot exists, then this operation does nothing.
<vm-path> reset <powerop_mode>	Shuts down, then reboots a virtual machine. The <code>powerop_mode</code> can be <code>hard</code> , <code>soft</code> , or <code>trysoft</code> . If <code>powerop_mode</code> is not specified, the default is <code>soft</code> . See “Values for <code>powerop_mode</code> ” on page 81.
<vm-path> revertstapshot	Reverts the virtual machine to the current snapshot. If no snapshot exists, then this operation does nothing, and the virtual machine state remains unchanged.
<vm-path> setguestinfo <variable> <value>	Writes a <code>GuestInfo</code> variable into memory. The variable is discarded when the virtual machine process terminates.
<vm-path> start <powerop_mode>	Powers on a previously powered-off virtual machine or resumes a suspended virtual machine. The <code>powerop_mode</code> can be <code>hard</code> , <code>soft</code> , or <code>trysoft</code> . If <code>powerop_mode</code> is not specified, the default is <code>soft</code> . See “Values for <code>powerop_mode</code> ” on page 81.
<vm-path> stop <powerop_mode>	Shuts down and powers off a virtual machine. The <code>powerop_mode</code> can be <code>hard</code> , <code>soft</code> , or <code>trysoft</code> . If <code>powerop_mode</code> is not specified, the default is <code>soft</code> . See “Values for <code>powerop_mode</code> ” on page 81.
<vm-path> suspend <powerop_mode>	Suspends a virtual machine. The <code>powerop_mode</code> can be <code>hard</code> , <code>soft</code> , or <code>trysoft</code> . If <code>powerop_mode</code> is not specified, the default is <code>soft</code> . See “Values for <code>powerop_mode</code> ” on page 81.

Values for `powerop_mode`

When you call `vmware-cmd` to start, stop, reset, or suspend a virtual machine, you must supply a value for the `powerop_mode` flag. This section describes what happens when that value is `soft`, `hard`, and `trysoft`.

NOTE You must have the current version of VMware Tools installed and running in the guest operating system to use a soft power operation.

Performing soft Power Operations

When you specify `soft` as the `powerop_mode` value, the command performs the following operations:

- **Stop** – Attempts to shut down the guest operating system, and then powers off the virtual machine.
- **Reset** – Attempts to shut down the guest operating system, and then reboots the virtual machine.
- **Suspend** – Attempts to run a script in the guest operating system before suspending the virtual machine.
- **Start** – The behavior depends on the state of the virtual machine.
 - **Start when a virtual machine is suspended** – After resuming the virtual machine, the command attempts to run a script in the guest operating system. The Start operation always succeeds. If VMware Tools is not present or is malfunctioning, running the script might fail.
 - **Start when virtual machine is powered off** – After powering on the virtual machine, the command attempts to run a script in the guest operating system when the VMware Tools service becomes active. The default script does nothing during this operation because it does not have to renew a DHCP lease. The Start operation always succeeds. If VMware Tools is not present or is malfunctioning, running the script might fail.

Performing hard Power Operations

When you specify `hard` as the `powerop_mode` value, the command performs the following operations:

- **Start** – Starts or resumes a virtual machine without running any scripts, that is, performs a standard power on or resume operation.
- **Stop, reset or suspend** – Immediately and unconditionally powers off, resets, or suspends the virtual machine.

Performing trysoft Power Operations

When you specify `trysoft` as the `powerop_mode` value, the system first attempts to perform the soft power transition operation. If this attempt fails, the system performs the hard power operation.

vmware-cmd Examples

The following examples for `vmware-cmd` require connection options. See [“Specifying Options for Remote CLI Commands”](#) on page 19.

```
vmware-cmd.pl <conn_params> C:\home\vmware\win2000.vmx getstate
```

Retrieves the execution state of a virtual machine. `C:\home\vmware\win2000.vmx` is the path to the virtual machine’s configuration file.

```
vmware-cmd.pl <conn_params> -q C:\home\vmware\win2000.vmx reset hard
```

Performs a hard reset, that is, the virtual machine is reset immediately and unconditionally.

- `-q` indicates the quiet option (only the results of the operation are printed).
- `C:\home\vmware\win2000.vmx` is the path to the virtual machine’s configuration file.

```
vmware-cmd.pl <conn_params> D:\dir.vmx connectdevice "CD/DVD Drive 2"
```

Connects the virtual IDE device CD/DVD Drive 2 in a Windows guest operating system.

```
vmware-cmd.pl <conn_params>  
/vmfs/volumes/4654bb7d-37601c39-9c29-00145e5a8a9d/padellvm2/padellvm2.vmx  
getproductinfo product
```

Returns, for example, `getproductinfo(product) = esx`.

Index

Numerics

2gbsparse disk format **70**

A

adding NAS datastore **26**
appliance See virtual appliance

B

backup of configuration information **45**

C

cloning **73**
cloning virtual disks **73**
command-line connection
 parameters **21**
commands with esxcfg prefix **61**
configuration files **20, 24**
configuration information **45**
connection options **19, 24**
copying files **59**
cp936 encoding **22**
creating directories **60**
creating pass-through RDM **76**
creating session files **19**
creating VMFS **68**

D

deleting virtual disks **72**
device target **66**
diagnostic partitions
 deactivating **29**
 managing **28**
 vicfg-dumppart **28**

directory groups **58**
disk attributes **69**
disk formats **69**
 supported **69**
disk partition target **66**
disks See virtual disks
displaying virtual disks **76**
DNS configuration **36**
downloading files **59**
downloading Remote CLI package **13**

E

eagerzeroedthick disk format **70**
editing remote files example **26**
encoding
 cp936 **22**
 ISO-8859-1 **22**
 Shift_JIS **22**
environment variables **20**
esxcfg prefix **61**
esxcfg-advcfg **55, 62**
esxcfg-cfgbackup **45, 62**
esxcfg-dns **36, 62**
esxcfg-dumppart **28, 62**
esxcfg-module **55, 62**
esxcfg-mpath **30, 62**
esxcfg-nas **33, 62**
esxcfg-nics **38, 62**
esxcfg-ntp **39, 62**
esxcfg-rescan **34, 62**
esxcfg-route **40, 62**
esxcfg-snmp **47, 62**

- esxcfg-syslog **49, 62**
 - esxcfg-user **56, 62**
 - esxcfg-vmhbadevs **35, 62**
 - esxcfg-vmknic **41, 62**
 - esxcfg-vswitch **42, 62**
 - examples
 - adding a NAS datastore **26**
 - cloning raw disk **73**
 - cloning virtual disk **73**
 - command line **25**
 - configuration files **24**
 - deleting a virtual disk **72**
 - editing files on ESX host **26**
 - editing remote files **26**
 - extending virtual disks **75**
 - extending VMFS volume **68**
 - inflating virtual disks **72**
 - initializing virtual disks **72**
 - listing attributes **69**
 - physical compatibility mode
 - RDM **76**
 - renaming virtual disks **73**
 - session files **24**
 - specifying connection options **24**
 - SSPI protocol **25**
 - svmotion **54**
 - vicfg-cfgbackup **46**
 - vicfg-dns **37**
 - vicfg-dumppart **29**
 - vicfg-module **55**
 - vicfg-mpath **32**
 - vicfg-nas **33**
 - vicfg-nics **38**
 - vicfg-ntp **40**
 - vicfg-rescan **34**
 - vicfg-route **40**
 - vicfg-snmp **48**
 - vicfg-syslog **49**
 - vicfg-user **57**
 - vicfg-vmhbadevs **35**
 - vicfg-vmknic **42**
 - vicfg-vswitch **44**
 - vifs **60**
 - virtual compatibility mode RDM **75**
 - virtual disks **71**
 - VMFS file system **68**
 - executing commands
 - Linux **13, 14**
 - Windows **15, 16**
 - execution options **22**
 - extending a virtual disk **74**
 - extending existing VMFS volume **68**
 - extending volume **68**
- ## F
- Fibre Channel storage **30**
 - file groups **58**
 - file system target **66**
 - file systems
 - NAS **33**
 - VMFS **68**
 - finding available LUNs **35**
 - fixed path policy **31**
- ## G
- groups **56**
 - GSX Server **74**
- ## H
- hard power operations **82**
 - host maintenance, vhostupdate **50**
- ## I
- importing virtual appliance **17**
 - inflating thin virtual disks **72**
 - initializing virtual disks **71**

- installing Remote CLI
 - Linux **13**
 - Windows **15**
- iSCSI storage **30**
- ISO-8859-1 encoding **22**
- L**
- Linux
 - executing commands **14**
 - installing Remote CLI **13**
 - Remote CLI **13**
 - shell **17**
- listing available LUNs **35**
- listing disk attributes **69**
- listing RDM attributes **76**
- listing VMFS volume attributes **69**
- logs **49**
- LUNs
 - listing available **35**
 - rescanning **34**
 - VML names **30**
- M**
- maintenance **50**
- managing diagnostic partitions **28**
- managing groups **56**
- managing NAS file systems **33**
- managing physical NICs **38**
- managing users **56**
- managing virtual switches **42**
- managing VMkernel NICs **41**
- manipulating route entry **40**
- Microsoft Windows Security Support
 - Provider Interface **21**
- migrating virtual machines
 - GSX Server **74**
 - svmotion **52**
 - vmkfstools **74**
 - VMware Workstation **74**
- miscellaneous management
 - commands **54**
- module options **55**
- monoflat disk format **70**
- monospaced disk format **70**
- most recently used path policy **31**
- MRU path policy **31**
- multipathing **30**
- N**
- NAS datastores
 - accessing **33**
 - adding in script **26**
 - adding to ESX host **33**
 - removing from ESX host **33**
 - vicfg-nas **26, 33**
- NetQueue configuration **55**
- network adapters
 - duplex value **38**
 - managing **38**
 - speed **38**
 - vicfg-nics **38**
 - vicfg-vmknic **41**
- Network Time Protocol **39**
- networking commands **36**
- NICs **38**
- NTP Server **39**
- O**
- options **22**
- P**
- parameters, command line **21**
- partitions, diagnostic **28**
- password, virtual appliance **18**
- path policies **31**
- performance monitoring **45**
- Perl **9**

- physical compatibility mode disk format **70**
- physical compatibility mode RDM **76**
- physical network adapters **38**
- physical NICs **38**
- port groups **42**
- power operations
 - hard **82**
 - soft **82**
 - trysoft **82**
- powerop_mode **81**

R

- raw device mapping
 - physical compatibility mode **76**
 - virtual compatibility mode **75**
- raw disks
 - cloning **73**
 - format **70**
 - rdm format **70**
 - rdmp format **70**
- RCLI *See* Remote CLI
- rdm disk format **70**
- rdmp disk format **70**
- RDMs
 - attributes **76**
 - physical compatibility mode **76**
 - virtual compatibility mode **75**
- Remote CLI
 - command-line **21**
 - configuration files **20**
 - environment variables **20**
 - execution options **22**
 - installing on Linux **13**
 - installing on Windows **15**
 - using virtual appliance **17**
- Remote CLI package
 - downloading **13**
 - installing on Linux **13**

- installing on Windows **15**
- uninstalling **15, 16**
- unpacking **13**
- Remote CLI virtual appliance
 - installing **17**
 - running **18**
- renaming virtual disks **73**
- required parameters **19**
- rescanning adapters **34**
- rescanning LUNs **34**
- resxtop **10, 45**
- root password **18**
- route entry **40**
- Running **59**
- running vifs **59**
- running virtual appliance **18**

S

- scripts with Remote CLI commands **25**
- session files **19, 24**
- Shift_JIS encoding **22**
- Simple Network Management Protocol **47**
- snapshots **74**
- soft power operations **82**
- SSPI protocol **21, 25**
- storage
 - commands **28**
 - creating directories with vifs **60**
- supported disk formats **69**
- svmotion **52**
- syslog server **49**
- system logs **49**
- system management commands **45**

T

- target, specifying **66**
- thick disk format **70**
- thin disk format **70**

- thin virtual disks
 - inflating **72**
 - vmkfstools **72**
- trysoft power operations **82**
- U**
- uninstalling Remote CLI package
 - Linux **15**
 - Windows **16**
- unpacking Remote CLI package **13**
- users, managing **56**
- using session files **19**
- V**
- variables **20**
- VI Perl Toolkit **9**
- vicfg-advcfg **55**
- vicfg-cfgbackup **45**
- vicfg-dns **36**
- vicfg-dumppart **28**
- vicfg-module
 - NetQueue configuration **55**
 - uses **55**
- vicfg-mpath **30**
- vicfg-nas **33**
- vicfg-nics **38**
- vicfg-ntp **39**
- vicfg-rescan **34**
- vicfg-route **40**
- vicfg-snmp **47**
- vicfg-syslog **49**
- vicfg-user **56**
- vicfg-vmhbadevs **35**
- vicfg-vmknic **41**
- vicfg-vswitch **42**
- vifs **58, 59**
- vihostupdate **50**
- virtual appliance
 - environment variables **20**
 - importing **17**
 - installing Remote CLI **17**
 - introduction **17**
 - multiple configuration files **20**
 - required parameters **19**
 - root password **18**
 - running **18**
 - using Remote CLI **17**
- virtual compatibility mode RDM **75**
- virtual disk geometry **76**
- virtual disks
 - cloning **73**
 - creating **70, 71**
 - deleting **72**
 - extending **75**
 - inflating **72**
 - initializing **71, 72**
 - options **69**
 - renaming **73**
- virtual machines
 - HBA names **30**
 - management commands **51**
 - managing **79**
 - migrating **52**
 - vmware-cmd **79**
- virtual switches
 - port groups **42**
 - vicfg-vswitch **42**
- VMFS
 - creating with vmkfstools **67, 68**
 - extending volumes **68**
 - listing attributes **69**
 - vmkfstools **63**
 - volume attributes **69**
- VMkernel module options **55**
- VMkernel NICs **41**
- vmkfstools
 - command syntax **64**
 - creating pass-through RDM **76**

- creating virtual disk **70**
- creating VMFS **67**
- creating VMFS example **68**
- deleting virtual disk **72**
- device target **66**
- disk formats **69**
- disk partition target **66**
- display disk geometry **76**
- extending virtual disk **74**
- extending volume **68**
- file system options **67**
- file system target **66**
- inflating thin virtual disk **72**
- initializing virtual disk **71**
- options **67**
- overview **63**
- RDM attributes **76**
- renaming virtual disk **73**
- snapshots **74**
- syntax **64**
- virtual disk options **69**
- VMFS volume attributes **69**

- VML LUN names **30**
- VMware Workstation **74**
- vmware-cmd
 - connection options **78**
 - examples **83**
 - general options **78**
 - server options **79**
 - virtual machine options **79**

W

- Windows

- executing commands **16**
- installing Remote CLI **15**
- using Remote CLI **15**

Z

- zeroedthick disk format **69**