

Getting Started with vSphere Command-Line Interfaces

ESXi 5.0
vCenter Server 5.0

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

EN-000488-00

vmware[®]

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

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

The VMware Web site also provides the latest product updates.

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

docfeedback@vmware.com

Copyright © 2010, 2011 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

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

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

Contents

About This Book	5
1 Managing vSphere with Command-Line Interfaces	7
Overview of vSphere Command-Line Interfaces	7
Using the vSphere Command-Line Interface	8
Using ESXCLI	8
ESXCLI Syntax	9
Running ESXCLI vCLI Commands	9
Running ESXCLI vCLI Commands Against ESXi 4.x Hosts	9
Running ESXCLI Commands in the ESXi Shell	9
ESXi Shell Access with the Direct Console	10
Enabling Local ESXi Shell Access	10
Using the Local ESXi Shell	11
Remote ESXi Shell Access with SSH	11
Enabling SSH for the ESXi Shell	11
Using the ESXi Shell with SSH	12
2 Installing vCLI	13
Installation Overview	13
Installing and Uninstalling vCLI on Linux	14
Installation Prerequisites	14
Installation Process	14
Installing Required Prerequisite Software for Your Linux Platform	16
Installing the vCLI Package	16
Uninstalling the vCLI Package on Linux	17
Installing and Uninstalling vCLI on Windows	17
Running Commands on Windows	18
Uninstalling the vCLI Package on Windows	18
Enabling Certificate Verification	19
Deploying vMA	19
3 Using the vSphere Command-Line Interface	21
Overview of Running Commands	21
Specifying Authentication Information	22
Order of Precedence for vCLI Authentication	22
Using a Session File	22
Using Environment Variables	23
Using a Configuration File	23
Using Command-Line Options	24
Using Microsoft Windows Security Support Provider Interface	25
vCLI and Lockdown Mode	25
Common Options for vCLI Execution	25
Using vCLI Commands in Scripts	27
ESXCLI Command Hierarchies	29
esxcli esxcli Namespace	29
esxcli fcoe Namespace	29

esxcli hardware Namespace	30
esxcli iscsi Namespace	30
esxcli license Namespace	30
esxcli network Namespace	31
esxcli software Namespace	31
esxcli storage Namespace	32
esxcli system Namespace	33
esxcli vm Namespace	33

About This Book

Getting Started with vSphere Command-Line Interfaces gives an overview of command-line interfaces in vSphere 5.0 and gets you started with ESXi Shell commands and vCLI (VMware® vSphere Command-Line Interface) commands. This book also includes instructions for installing vCLI and a reference to connection parameters.

Intended Audience

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

VMware Technical Publications Glossary

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

Document Feedback

VMware welcomes your suggestions for improving our documentation. If you have comments, send your feedback to docfeedback@vmware.com.

Related Documentation

The documentation for vCLI is available in the vSphere Documentation Center and on the vCLI documentation page. Go to <http://www.vmware.com/support/pubs>, select **VMware Administration Products**, and select **vSphere Command-Line Interface**.

- *Command-Line Management in vSphere 5.0 for Service Console Users* is a technical note for users who are currently using ESX service console commands, scripts, agents, or logs. You learn how to transition to an off-host implementation or to use the ESXi Shell in special cases.
- *vSphere Command-Line Interface Concepts and Examples* presents usage examples for many commands, such as setting up software and hardware iSCSI, adding virtual switches, setting up Active Directory authentication, and so on. The document includes the same example with the `vicfg-` command and the ESXCLI command where supported.
- *vSphere Command-Line Interface Reference* is a reference to both `vicfg-` commands and ESXCLI commands. The `vicfg-` command help is generated from the POD available for each command, run `pod2html` for any `vicfg-` command to generate individual HTML files interactively. The ESXCLI reference information is generated from the ESXCLI help.

The documentation for PowerCLI is available in the vSphere Documentation Center and on the PowerCLI documentation page. Go to <http://www.vmware.com/support/pubs>, select **VMware Administration Products**, and select **vSphere PowerCLI documentation**.

The vSphere SDK for Perl documentation explains how you can use the vSphere SDK for Perl and related utility applications to manage your vSphere environment. The documentation includes information about the vSphere SDK for Perl Utility Applications.

The *vSphere Management Assistant Guide* explains how to install and use the vSphere Management Assistant (vMA). vMA is a virtual machine that includes vCLI and other prepackaged software. See “[Deploying vMA](#)” on page 19.

Background information for the tasks discussed in this book is available in the vSphere documentation set. The vSphere documentation consists of the combined VMware vCenter Server and ESXi documentation.

Technical Support and Education Resources

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

Online and Telephone Support

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

Customers with appropriate support contracts should use telephone support for the fastest response on priority 1 issues. Go to http://www.vmware.com/support/phone_support.

Support Offerings

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

VMware Professional Services

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

Managing vSphere with Command-Line Interfaces

1

vSphere supports several command-line interfaces for managing your virtual infrastructure including the vSphere Command-Line Interface (vCLI), a set of ESXi Shell commands, and PowerCLI. You can choose the CLI set best suited for your needs, and write scripts to automate your CLI tasks.

This chapter includes the following topics:

- [“Overview of vSphere Command-Line Interfaces”](#) on page 7
- [“Using the vSphere Command-Line Interface”](#) on page 8
- [“Using ESXCLI”](#) on page 8
- [“ESXi Shell Access with the Direct Console”](#) on page 10
- [“Remote ESXi Shell Access with SSH”](#) on page 11

Overview of vSphere Command-Line Interfaces

vSphere includes commands for managing different aspects of your environment, either locally or remotely.

Command set	Description	See
ESXCLI commands	<p>Manage many aspects of an ESXi host. You can run ESXCLI commands remotely or in the ESXi Shell.</p> <ul style="list-style-type: none">■ vCLI package. Install the vCLI package on the server of your choice, or deploy a vMA virtual machine and target the ESXi system that you want manipulate. You can run ESXCLI commands against a vCenter Server system and target the host indirectly. Running against vCenter Server systems by using the <code>-vihost</code> parameter is required if the host is in lockdown mode.■ ESXi shell. Run ESXCLI commands in the local ESXi shell to manage that host. <p>You can also run ESXCLI commands from the vSphere PowerCLI prompt by using the <code>Get-ESXcli cmdlet</code>.</p>	<p>“Using ESXCLI” on page 8</p> <p>“Installing vCLI” on page 13</p> <p><i>vSphere Command-Line Concepts and Examples</i></p> <p><i>vSphere Management Assistant Guide</i></p> <p><i>vSphere Command-Line Interface Reference</i></p>
esxcfg- commands	<p>Available in the ESXi Shell. <code>esxcfg-</code> commands are still included in this release but are deprecated. Migrate to ESXCLI where possible. ESXCLI replacements exist for most commands.</p>	<p><i>Command-Line Management of vSphere 5.0 for Service Console Users</i></p>
vicfg- and other vCLI commands	<p>Introduced in vSphere 3 to allow users to manage hosts remotely. Install the vCLI package on the server of your choice, or deploy a vMA virtual machine and target the ESXi system that you want manipulate.</p> <p>You can run the commands against ESXi systems or against a vCenter Server system. If you target a vCenter Server system, use the <code>--vihost</code> option to specify the target ESXi system.</p> <p>Note: If the ESXi system is in lockdown mode, you must run commands against the vCenter Server system that manages your ESXi system.</p>	<p>“Installing vCLI” on page 13</p> <p><i>vSphere Command-Line Concepts and Examples</i></p> <p><i>vSphere Command-Line Interface Reference</i></p>

Command set	Description	See
VMware PowerCLI cmdlets	VMware vSphere PowerCLI provides a Windows PowerShell interface to the vSphere API. vSphere PowerCLI includes PowerShell cmdlets for administering vSphere components. In addition, the vSphere PowerCLI package includes the vSphere SDK for .NET for developers who want to create their own applications. vSphere PowerCLI includes more than 200 cmdlets, a set of sample scripts, and a function library for management and automation. The vSphere Image Builder PowerCLI and the vSphere Auto Deploy PowerCLI are included when you install the vSphere PowerCLI.	VMware PowerCLI documentation set.
localcli commands	Set of commands for use with VMware Technical Support. localcli commands are equivalent to ESXCLI commands, but bypass hostd. The localcli commands are only for situations when hostd is unavailable and cannot be restarted. After you run a localcli command, you must restart hostd. Run ESXCLI commands after the restart. If you use a localcli command in other situations, an inconsistent system state and potential failure can result.	

Using the vSphere Command-Line Interface

The vCLI command set includes vicfg- commands and ESXCLI commands. The ESXCLI commands included in the vCLI package are equivalent to the ESXCLI commands available on the ESXi Shell. The vicfg- command set is similar to the deprecated esxcfg- command set in the ESXi Shell.

IMPORTANT ESXi Shell is intended for experienced users only. Minor errors in the shell can result in serious problems. Instead of running commands directly in the ESXi Shell, use vCLI or PowerCLI.

You can run vCLI commands from a Windows or Linux system, or use vMA.

- Install the vCLI command set on the Windows or Linux system from which you want to administer your ESXi systems and run vCLI commands. See [“Installing vCLI”](#) on page 13.
- Deploy a vMA virtual machine to an ESXi system and run vCLI commands from there.

After you have installed the vCLI package you can run the commands in the set against ESXi hosts. You must specify connection parameters when you run a vCLI command. See [“Using the vSphere Command-Line Interface”](#) on page 21.

Using ESXCLI

You can manage many aspects of an ESXi host with the ESXCLI command set. You can run ESXCLI commands as vCLI commands or run them in the ESXi Shell in troubleshooting situations.

You can also run ESXCLI commands from the PowerCLI shell by using the Get-ESxcli cmdlet. See the *vSphere PowerCLI Administration Guide* and the *vSphere PowerCLI Reference*.

The set of ESXCLI commands available on a host depends on the host configuration. The *vSphere Command-Line Interface Reference* lists help information for all ESXCLI commands. Run `esxcli --server <MyESXi> --help` before you run a command on a host to verify that the command is defined on the host you are targeting.

ESXCLI Syntax

Each ESXCLI 5.0 command uses the same syntax.

```
esxcli [dispatcher options] <namespace> [<namespace> ...] <cmd> [cmd options]
```

- **dispatcher options.** Predefined options for connection information such as target host, user name, and so on. See [“Common Options for vCLI Execution”](#) on page 25. Not required when you run the command in the ESXi Shell.
- **namespace.** Groups ESXCLI commands. vSphere 5.0 supports nested namespaces.
- **command.** Reports on or modifies state on the system.

Running ESXCLI vCLI Commands

You can run an ESXCLI vCLI command remotely against a specific host or against a vCenter Server system. You have the following choices:

- Deploy the vMA appliance on an ESXi system and authenticate against a set of target servers. You can then run ESXCLI commands against any target server by specifying the `--host` dispatcher option. No additional authentication is required. See the *vSphere Management Assistant Guide*.
- Install the vCLI package on one of the supported Windows or Linux systems. The ESXCLI command set is included. You can run commands against an ESXi or vCenter Server system if you specify connection options. See [“Installing vCLI”](#) on page 13.

After installation, run ESXCLI commands against a specific host by first specifying all dispatcher options. If the target server is a vCenter Server system, specify the target ESXi host before any ESXCLI namespaces, commands, and supported options.

```
esxcli --server myESXi --username user1 --password 'my_password' storage nfs list
esxcli --server myVCServer --username user1 --password 'my_pwd' --vihost myESXi.mycompany.com
storage nfs list
```

Each time you run a command, you must specify authentication information. See [“Using the vSphere Command-Line Interface”](#) on page 21.

Running ESXCLI vCLI Commands Against ESXi 4.x Hosts

When you run an ESXCLI vCLI command, you must know the commands supported on the target host specified with `--server` or as a vMA target.

- If you run commands against ESXi 4.x hosts, ESXCLI 4.x commands are supported.
- If you run commands against ESXi 5.0 hosts, ESXCLI 5.0 commands are supported.

VMware partners might develop custom ESXCLI commands that you can run on hosts where the partner VIB has been installed.

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

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

Running ESXCLI Commands in the ESXi Shell

ESXCLI commands in the ESXi Shell are fully supported unless they are marked as internal in the online help.

The ESXi Shell is disabled by default. You must enable the ESXi Shell before you can run commands in the shell. See [“ESXi Shell Access with the Direct Console”](#) on page 10.

To run an ESXCLI command in the shell

- 1 Log in to the shell.
- 2 Run the command. For example, to list NAS storage devices, run the following command.

```
esxcli storage nfs list
```

You can use `--help` at any level of `esxcli` for help on available namespaces, commands, or options.

ESXi Shell Access with the Direct Console

An ESXi system includes a direct console (also called DCUI) that allows you to start and stop the system and to perform a limited set of maintenance and troubleshooting tasks. The direct console includes the ESXi Shell, which is disabled by default. You can enable the ESXi Shell in the direct console or by using the vSphere Client. You can enable local shell access or remote shell access:

- Local shell access allows you to log in to the shell directly from the Direct Console. See [“Enabling Local ESXi Shell Access”](#) on page 10.
- Remote shell (SSH) access allows you to connect to the host using a shell such as PuTTY, specify a user name and password, and run commands in the shell.

The ESXi Shell includes all ESXCLI commands, a set of deprecated `esxcfg-` commands, and a set of commands for troubleshooting and remediation.

IMPORTANT All ESXCLI commands that are available in the ESXi Shell are also included in the vCLI package.

VMware recommends you install the vCLI package on a supported Windows or Linux system or deploy the vMA virtual appliance, and run commands against your ESXi hosts. Run commands directly in the ESXi Shell in troubleshooting situations only.

Enabling Local ESXi Shell Access

You can enable the ESXi Shell from the direct console or from the vSphere Client.

If you have access to the direct console, you can enable the ESXi Shell from there.

To enable the ESXi Shell in the direct console

- 1 At the direct console of the ESXi host, press F2 and provide credentials when prompted.
- 2 Scroll to **Troubleshooting Options** and press Enter.
- 3 Choose **Enable ESXi Shell** and press Enter.

On the left, **Enable ESXi Shell** changes to **Disable ESXi Shell**. On the right, **ESXi Shell is Disabled** changes to **ESXi Shell is Enabled**.

- 4 Press Esc until you return to the main direct console screen.

If you do not have access to the direct console, you can enable the ESXi Shell from the vSphere Client.

To enable the local or remote ESXi Shell from the vSphere Client

- 1 Select the host, click the **Configuration** tab, and click **Security Profile** in the Software panel.
- 2 In the Services section, click **Properties**.
- 3 Select **ESXi Shell** and click **Options**.
- 4 Change the ESXi Shell options.
 - To change the Startup policy across reboots, click **Start and stop with host** and reboot the host.
 - To temporarily start or stop the service, click the **Start** or **Stop** button.
- 5 Click **OK**.

After you have enabled the ESXi Shell, you can use it from that monitor or through a serial port.

The ESXi Shell timeout setting specifies how long, in minutes, you can leave an unused session open. By default, the timeout for the ESXi Shell is 0, which means the session remains open even if it is unused. If you change the timeout, for example, to 30 minutes, you have to log in again after the timeout period has elapsed.

NOTE If you are logged in when the timeout period elapses, your session will persist. However, the ESXi Shell will be disabled, preventing other users from logging in.

You can modify the timeout from the Direct Console or from the vSphere Client.

To modify the ESXi Shell Timeout

- In the Direct Console, follow these steps.
 - a Select **Modify ESXi Shell timeout** and press Enter.
 - b Enter the time-out value in minutes and press Enter.
- In the vSphere Client, follow these steps.
 - a In the Configuration tab's Software panel, click **Advanced Settings**.
 - b In the left panel, click **UserVars**.
 - c Locate **UserVars.ESXiShellTimeOut** and enter the timeout value in minutes.
 - d Click **OK**.

Using the Local ESXi Shell

After you enable the ESXi Shell in the direct console, you can use it from main direct console screen or remotely through a serial port.

To use the local ESXi Shell

- 1 At the main direct console screen, press Alt-F1 to open a virtual console window to the host.
- 2 Provide credentials when prompted.

When you type the password, characters are not displayed on the console.
- 3 Enter shell commands to perform management tasks.
- 4 To log out, type `exit` in the shell.
- 5 To return to the direct console, type Alt-F2.

See vSphere *Installation and Setup* documentation for information on serial port setup.

Remote ESXi Shell Access with SSH

If Secure Shell is enabled for the ESXi Shell, you can run shell commands by using a Secure Shell client such as SSH or PuTTY.

Enabling SSH for the ESXi Shell

By default, remote command execution is disabled on an ESXi host, and you cannot log in to the host using a remote shell. You can enable remote command execution from the direct console or from the vSphere Client.

To enable SSH access in the direct console

- 1 At the direct console of the ESXi host, press F2 and provide credentials when prompted.
- 2 Scroll to **Troubleshooting Options** and press Enter.

- 3 Choose **Enable SSH** and press Enter once.

On the left, **Enable SSH** changes to **Disable SSH**. On the right, **SSH is Disabled** changes to **SSH is Enabled**.

- 4 Press Esc until you return to the main direct console screen.

To enable SSH from the vSphere Client

- 1 Select the host and click the **Configuration** tab.
- 2 Click **Security Profile** in the Software panel.
- 3 In the Services section, click **Properties**.
- 4 Select **SSH** and click **Options**.
- 5 Change the SSH options.
 - To change the Startup policy across reboots, click **Start and stop with host** and reboot the host.
 - To temporarily start or stop the service, click the **Start** or **Stop** button.
- 6 Click **OK**.

After you have enabled SSH, you log in to the ESXi Shell remotely and run ESXi Shell commands.

Using the ESXi Shell with SSH

If SSH is enabled on your ESXi host, you can run commands on that shell using an SSH client.

To access the remote ESXi Shell

- 1 Open an SSH client.
- 2 Specify the IP address or domain name of the ESXi host.

Precise directions vary depending on the SSH client that you are using. See vendor documentation and support.
- 3 Provide credentials when prompted.

Installing vCLI

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

This chapter includes the following topics:

- [“Installation Overview”](#) on page 13
- [“Installing and Uninstalling vCLI on Linux”](#) on page 14
- [“Installing and Uninstalling vCLI on Windows”](#) on page 17
- [“Enabling Certificate Verification”](#) on page 19
- [“Deploying vMA”](#) on page 19

Installation Overview

You can install a vCLI package on a supported platform of your choice or deploy the vMA virtual machine on an ESXi host.

- **vCLI packages.** Install a vCLI package on a physical or virtual machine. See [“Installing and Uninstalling vCLI on Linux”](#) on page 14 and [“Installing and Uninstalling vCLI on Windows”](#) on page 17.

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

After you have installed the package, you can run vCLI commands and vSphere SDK for Perl utility applications from the operating system command line. Each time you run a command, you specify the target server connection options directly or indirectly. See [“Specifying Authentication Information”](#) on page 22. You can also write scripts and manage your vSphere environment using those scripts.

- **vMA.** Deploy vMA, a virtual machine that administrators can use to run scripts that manage vSphere, on an ESXi host. vMA includes vCLI, vSphere SDK for Perl, and other prepackaged software in a Linux environment.

vMA supports noninteractive login. If you establish an ESXi host as a target server, you can run vCLI and vSphere SDK for Perl commands against that server without additional authentication. If you establish a vCenter Server system as a target server, you can run most vCLI commands against all ESXi systems that server manages without additional authentication. See [“Deploying vMA”](#) on page 19.

Installing and Uninstalling vCLI on Linux

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

- Red Hat Enterprise Linux 5.5 Server (32 bit and 64 bit)
- SLES 10 SP1 (32 bit and 64 bit)
- SLES 11 (32 bit and 64 bit)
- Ubuntu 10.04 (32 bit and 64 bit)

The Linux installer for vCLI 5.0.0 stops if you do not have required software installed. The installer uses CPAN to download and install other prerequisite Perl modules from CPAN sites. You must have an Internet connection to allow the installer to download and install Perl modules from CPAN.

Install vCLI on Linux only if you are an experienced Linux administrator who knows how to use the system's package manager. Otherwise, use vMA. See [“Deploying vMA”](#) on page 19.

Installation Prerequisites

Install the vCLI package on a Linux system that meets the following prerequisites.

- **Internet access.** You must have Internet access when you run the installer because the installer uses CPAN to install prerequisite Perl modules.
- **Development Tools and Libraries.** You must install the Development Tools and Libraries for the Linux platform that you are working with before you install vCLI and prerequisite Perl modules.
- **Proxy settings.** If your system is using a proxy for Internet access, you must set the `http://` and `ftp://` proxies, as follows:

```
export http_proxy=<proxy_server>:port
export ftp_proxy=<proxy_server>:port
```

Installation Process

The vCLI package installer installs the vCLI scripts and the vSphere SDK for Perl. The installation proceeds as follows.

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

Perl	Perl version 5.8.8 or version 5.10 must be installed on your system.
OpenSSL	The vCLI requires SSL because most connections between the system on which you run the command and the target vSphere system are encrypted with SSL. The OpenSSL library (<code>libssl-dev</code> package) is not included in the default Linux distribution. See “Installing Required Prerequisite Software for Your Linux Platform” on page 16.
LibXML2	Used for XML parsing. The vCLI client requires 2.6.26 or higher version. If you have an older version installed, please upgrade to 2.6.26 or higher. The <code>libxml2</code> package is not included in the default Linux distribution. “Installing Required Prerequisite Software for Your Linux Platform” on page 16.
uuid	Included in <code>uuid-dev</code> for SLES 11 and in <code>e2fsprogs</code> for other Linux platforms. Required by the UUID Perl module.

- 2 If the required software is found, the installer proceeds. Otherwise, the installer stops and informs you that you must install the software. See [“Installing Required Prerequisite Software for Your Linux Platform”](#) on page 16 for instructions.
- 3 The installer checks whether the following Perl modules are found, and whether the correct version is installed.
 - Crypt-SSLeay-0.55 (0.55-0.9.7 or 0.55-0.9.8)
 - IO-Compress-Base-2.005
 - Compress-Zlib-2.005

- IO-Compress-Zlib-2.005
- Compress-Raw-Zlib-2.017
- Archive-Zip-1.26
- Data-Dumper-2.121
- XML-LibXML-1.63
- libwww-perl-5.805
- LWP-Protocol-https-6.02
- XML-LibXML-Common-0.13
- XML-Namespacesupport-1.09
- XML-SAX-0.16
- Data-Dump-1.15
- URI-1.37
- UUID-0.02
- SOAP-Lite-0.710.08
- HTML-Parser-3.60
- version-0.78

Earlier versions of libwww-perl include the LWP-Protocol-https module. Very recent versions of libwww-perl do not include the LWP-Protocol-https module.

- 4 The installer proceeds depending on whether the Perl modules are found.
 - If a recommended Perl module is not found at all, the installer installs it using CPAN. You must meet the installation prerequisites or the installer cannot install the Perl modules and stops. See [“Installation Prerequisites”](#) on page 14.
 - If a lower version of a recommended module is found, the installer does not install a different version from CPAN and proceeds with installation. After completing installation, the installer displays a message that the version on the system does not match the recommended version, and recommends that you install the version vCLI was tested with. You can install the modules using the package installer for your platform, the installation CD, or CPAN.
 - If a higher version of a recommended module is found, the installer proceeds with installation and does not display a message after installation.

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

- 5 After all required software and all prerequisite Perl modules are installed, you can install vCLI. See [“Installing the vCLI Package”](#) on page 16.

If a previous version of vCLI, Remote CLI, or vSphere SDK for Perl is installed on your system, and you install vCLI in a different directory, you must reset the PATH environment variable. You can do so before or after the installation, using the command appropriate for your distribution and shell (`setenv`, `export`, and so on). If you do not reset the path, the system might still look for executables in the old location.

Installing Required Prerequisite Software for Your Linux Platform

If required prerequisite software is not installed, the installer stops and requests that you install it. Installation depends on the platform that you are using.

Table 2-1. Installing Required Prerequisite Software

Platform	Installation
Red Hat Enterprise Linux 5.5, 32-bit and 64-bit	Install prerequisites using yum, the RHEL package installer (recommended), or from the installation DVD. For example: <pre>yum install openssl-devel libxml2-dev e2fsprogs-dev</pre>
SLES 10, 32 bit SLES 10, 64 bit	Install the prerequisite packages from the SLES 10 and SLES 11 SDK DVD. When you insert the DVD, it offers to auto run. Cancel the auto run dialog box and use the yast package installer to install OpenSSL or other missing required packages. <ul style="list-style-type: none"> ■ SLES 10, 64 bit. <code>yast -i openssl-devel libxml2-devel-32bit e2fsprogs-devel-32bit</code> ■ SLES 10, 32 bit. <code>yast -i openssl-devel libxml2-devel e2fsprogs-devel</code> Some users might be authorized to use the Novell Customer Center and use yast to retrieve missing packages from there. Note that SLES 10 includes libxml2 version 2.6.23. The vCLI client require 2.6.26 or higher. Upgrade to 2.6.26 or higher.
SLES 11, 32 bit SLES 11, 64 bit	Install the prerequisite packages from the SLES 10 and SLES 11 SDK DVD. When you insert the DVD, it offers to auto run. Cancel the auto run dialog box and use the yast package installer to install OpenSSL or other missing required packages. <ul style="list-style-type: none"> ■ SLES 11 64 bit. <code>yast -i openssl-devel libuuid-devel libuuid-devel-32bit</code> ■ SLES 11 32 bit. <code>yast -i openssl-devel libuuid-devel</code> Some users might be authorized to use the Novell Customer Center and use yast to retrieve missing packages from there.
Ubuntu 10.04, 32 bit Ubuntu 10.04, 64 bit	<ol style="list-style-type: none"> 1. Connect to the Internet. 2. Update the local repository of libraries from a terminal window. <pre>sudo apt-get update</pre> 3. Install the required libraries from a terminal window. <ul style="list-style-type: none"> ■ 32 bit. <code>sudo apt-get install build-essential gcc uuid uuid-dev perl libssl-dev perl-doc liburi-perl libxml-libxml-perl libcrypt-ssleay-perl</code> ■ 64 bit. <code>sudo apt-get install ia32-libs build-essential gcc uuid uuid-dev perl libssl-dev perl-doc liburi-perl libxml-libxml-perl libcrypt-ssleay-perl</code> For Ubuntu 10.04 64 bit, you must install the 32-bit compatibility libraries or the <code>resxtop</code> and <code>ESXCLI</code> commands do not work.

Installing the vCLI Package

Install the vCLI package and run a command to verify installation was successful.

To install vCLI on Linux

- 1 Log in as root.
- 2 Untar the vCLI binary that you downloaded.

```
tar -zxvf VMware-vSphere-CLI-5.X.X-XXXXX.i386.tar.gz
```

 A `vmware-vsphere-vcli-distrib` directory is created.
- 3 (Optional) If your server uses a proxy to access the Internet, and if your `http://` and `ftp://` proxy were not set when you installed prerequisite software, set them now.

```
export http_proxy=<proxy_server>:port
export ftp_proxy=<proxy_server>:port
```
- 4 Run the installer:

```
/sudo vmware-vsphere-cli-distrib/vmware-install.pl
```
- 5 To accept the license terms, type **yes** and press Enter.

The installer connects to CPAN and installs prerequisite software. This might take a long time.

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

A complete installation process has the following result:

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

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

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

See the vSphere SDK for Perl documentation for a reference to all utility applications. After you install vCLI, you can test the installation by running a vCLI command or vSphere SDK for Perl utility application from the command prompt.

To run a vCLI command on Linux

- 1 Open a command prompt.
- 2 (Optional) Change to the directory where you installed the vCLI (default is `/usr/bin`).
- 3 Run the command, including the connection options.

```
<command> <conn_options> <params>
```

Specify connection options in a configuration file or pass them on the command line. The extension `.pl` is not required on Linux. For example:

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

The system prompts you for a user name and password for the target server.

See [Table 3-2, “vCLI Connection Options,”](#) on page 26 for a complete list of connection options.

Uninstalling the vCLI Package on Linux

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

To uninstall vCLI on Linux

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

The command uninstalls vCLI and the vSphere SDK for Perl.

Installing and Uninstalling vCLI on Windows

Before you can run vCLI commands from your Windows system, you must install the vCLI package and test the installation by running a command.

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

- Windows Vista Enterprise SP1 32 bit and 64 bit
- Windows 2008 64 bit
- Windows 7 32 bit and 64 bit

IMPORTANT If you want to run ESXCLI commands included in vCLI from a Windows system, you must have the Visual C++ 2008 redistributable for 32 bit installed on that system. Find `vc_redist_x86.exe` for Visual C++ 2008 and install it on your Windows system.

To install the vCLI Package on Windows

- 1 Download the vCLI Windows installer package.
You can find the installer on the VMware Communities page.
- 2 Start the installer.
- 3 (Optional) If prompted to remove older versions of vSphere SDK for Perl or vCLI, you can either accept or cancel the installation and install the vCLI package on a different system.

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

- 4 Click **Next** in the Welcome page.
- 5 To install the vCLI in a nondefault directory, click **Change** and select the directory.
The default location is C:\Program Files\VMware\VMware vSphere CLI.
- 6 Click **Next**.
- 7 Click **Install** to proceed with the installation.
The installation might take several minutes to complete.
- 8 Reboot your system.
Without reboot, path settings might not be correct on your Windows platform.

Running Commands on Windows

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

To run a vCLI command on Windows

- 1 From the Windows Start menu, choose **Programs > VMware > VMware vSphere CLI > Command Prompt**.
A command prompt shell for the location where vCLI is installed appears. You have easy access to vCLI and to vSphere SDK for Perl commands from that location.
- 2 Run the command, passing in connection options and other options.
On Windows, the extension `.pl` is required for `vicfg-` commands, but not for `ESXCLI`.
`<command>.pl <conn_options> <params>`
For example:
`vicfg-nas.pl --server my_esxhost --list`
The system prompts you for a user name and password.

See [Table 3-2, “vCLI Connection Options,”](#) on page 26 for a complete list of connection options.

Uninstalling the vCLI Package on Windows

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

To uninstall vCLI on Windows

- 1 Find the option for adding and removing programs on the Windows operating system you are using.
- 2 In the panel that appears, select **VMware vSphere CLI**, and click **Remove**.
- 3 Click **Yes** when prompted.

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

Enabling Certificate Verification

The vSphere SDK for Perl and vCLI use `Crypt::SSLEay` to support certificate verification. `Crypt::SSLEay` allows verification of certificates signed by a Certificate Authority (CA) if you set the following two variables:

- `HTTPS_CA_FILE` – The CA file.
- `HTTPS_CA_DIR` – The CA directory.

See the `Crypt::SSLEay` documentation for details on setup.



CAUTION If the two environment variables `HTTPS_CA_FILE` and `HTTPS_CA_DIR` are set incorrectly or if a problem with the certificate exists, vCLI commands do not complete, and do not print error or warning messages. Use `HTTPS_DEBUG` for troubleshooting before running vCLI commands.

Deploying vMA

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

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

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

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

- 2 Configure vMA.

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

- Network information (the default is often acceptable)
 - Host name for vMA.
 - Password for the vi-admin user. The vi-admin user has superuser privileges on vMA. You cannot log in to vMA as the root user.
- 3 (Optional) Add a vCenter Server system or one or more ESXi systems as targets. You configure vMA for Active Directory authentication and can then add ESXi and vCenter Server systems to vMA without having to store passwords in the vMA credential store. See the *vSphere Management Assistant Guide*.

Using the vSphere Command-Line Interface

3

You can run vSphere Command-Line Interface (vCLI) commands from the command line of the system where you installed the package, from the vMA command line, and from scripts. Each command requires at a minimum the target server to run the command on. Users authorized to run commands on the target server do not have to specify authentication information. Other users must specify authentication information.

This chapter includes the following topics:

- [“Overview of Running Commands”](#) on page 21
- [“Specifying Authentication Information”](#) on page 22
- [“Common Options for vCLI Execution”](#) on page 25
- [“Using vCLI Commands in Scripts”](#) on page 27

IMPORTANT If an ESXi system that you target is in lockdown mode, you cannot run vCLI commands against that system directly. You must target a vCenter Server system that manages the ESXi system and use the `--vhost` option to specify the ESXi target. See [“vCLI and Lockdown Mode”](#) on page 25.

Overview of Running Commands

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

- Open a command prompt on a Linux or Windows system on which you installed vCLI. Enter commands into that command prompt, specifying connection options.
- Access the vMA Linux console. Set up target servers and run vCLI commands against the targets without additional authentication.
- Prepare scripts that contain vCLI commands. Then run the scripts from a remote administration server that has the vCLI package installed or from the vMA Linux console. See [“Using vCLI Commands in Scripts”](#) on page 27.

When you run commands against an ESXi host, you must be authenticated for that host. When you run commands against a vCenter Server system, and you are authenticated for that system, you can target all ESXi hosts that vCenter Server manages without additional authentication. See [“Specifying Authentication Information”](#) on page 22.



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

Follow one of the following approaches for protecting passwords.

- If you use a vCLI command interactively and do not specify a user name and password, you are prompted for them. The screen does not echo the password you type.
- For noninteractive use, you can create a session file using the `save_session` script included in the `apps/session` directory. See [“Using a Session File”](#) on page 22.
- If you are running on a Windows system, you can use the `--passthroughauth` option. If the user who runs the command with that option is known, no password is required.

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

Specifying Authentication Information

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

IMPORTANT vCLI 4.1 and later allows administrators to place ESXi hosts in lockdown mode for enhanced security. Only a vCLI command or a vSphere Client connected to a vCenter Server system can make changes to ESXi hosts in lockdown mode. No users, not even the root user, can run vCLI commands against ESXi hosts in lockdown mode. See [“vCLI and Lockdown Mode”](#) on page 25 and the *Datacenter Administration Guide*.

Order of Precedence for vCLI Authentication

When you run a vCLI command, authentication happens in the order of precedence shown in [Table 3-1](#). This order of precedence always applies. That means, for example, that you cannot override an environment variable setting in a configuration file.

Table 3-1. vCLI Authentication Precedence

Authentication	Description	See
Command line	Password (<code>--password</code>), session file (<code>--sessionfile</code>), or configuration file (<code>--config</code>) specified on the command line.	“Using a Session File” on page 22
Environment variable	Password specified in an environment variable.	“Using Environment Variables” on page 23
Configuration file	Password specified in a configuration file.	“Using a Configuration File” on page 23
Current account (Active Directory)	Current account information used to establish an SSPI connection. Available only on Windows.	“Using Microsoft Windows Security Support Provider Interface” on page 25
Credential store	Password retrieved from the credential store.	<i>vSphere Web Services SDK Programming Guide</i> and <i>vSphere SDK for Perl Programming Guide</i> .
Prompt the user for a password.	Password is not echoed to screen.	

Using a Session File

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

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

To create and use a session file

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

For example:

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

Linux: `cd /usr/share/lib/vmware-vccli/apps/session`

- 2 Run `save_session`.

You can use the `save_session.pl` script or the `--savesessionfile` option to the vCLI command. You must specify the server to connect to and the name of a session file in which the script saves an authentication cookie.

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

For example:

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

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

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

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

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

For example:

Windows: `esxcli --sessionfile C:\Temp\my_session network ip interface list vicfg-mpath.pl --sessionfile C:\Temp\my_session --list`

Linux: `esxcli --sessionfile /tmp/vimsession network ip interface list vicfg-mpath --sessionfile /tmp/vimsession --list`

Using Environment Variables

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

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

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

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

IMPORTANT Do not use escape characters in environment variables.

See [“Using vCLI Commands in Scripts”](#) on page 27 for an environment variable example.

Using a Configuration File

You can use a text file that contains variable names and settings as a configuration file. Variables corresponding to the options are shown in [Table 3-2, “vCLI Connection Options,”](#) on page 26.



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

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

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

For example:

```
esxcli --config <my_saved_config> network ip interface list
vicfg-mpath --config <my_saved_config> --list
```

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

The following example illustrates the contents of a configuration file:

```
VI_SERVER = XX.XXX.XXX.XX
VI_USERNAME = root
VI_PASSWORD = my_password
VI_PROTOCOL = https
VI_PORTNUMBER = 443
```

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

Using Command-Line Options

You can pass in command-line options using option name and option value pairs in most cases. For ESXCLI commands, you can use long or short options. An equal sign between option name and option value is optional.

```
esxcli --server <vc_server> --username <privileged_user> --password <pw> --vihost <esx_host>
      <namespace> [<namespace>]... <command> --<option_name=option_value>
```

For other vCLI commands, use long or short options. An equal sign is not supported.

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

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

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

When running commands on Windows, use double quotes (“ ”). When running commands on Linux, use single quotes (‘ ’) or a backslash (\) as an escape character.

The following examples connect to the server as user `snow-white` with password `dwarf$`.

Linux

```
esxcli --server <server> --username snow\white --password dwarf\$ network ip interface list
esxcli --server <server> --username snow\white --password 'dwarf$' network ip interface list
vicfg-mpath --server <server> --username snow\white --password dwarf\$ --list
vicfg-mpath --server <server> --username 'snow-white' --password 'dwarf$' --list
```

Windows

```
esxcli --server <server> --username "snow-white" --password "dwarf$" network ip interface list
vicfg-mpath.pl --server <server> --username "snow-white" --password "dwarf$" --list
```


Using Microsoft Windows Security Support Provider Interface

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

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

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

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

Example

```
esxcli --server <vc_server> --passthroughauth --passthroughauthpackage "Kerberos"
      --vihost my_esx network ip interface list
```

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

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

vCLI and Lockdown Mode

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

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

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

Common Options for vCLI Execution

Table 3-2 lists options that are available for all vCLI commands in alphabetical order. The table includes options for use on the command line and variables for use in configuration files.

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

See [“To run a vCLI command on Linux”](#) on page 17 and [“To run a vCLI command on Windows”](#) on page 18 for usage examples.

Table 3-2. vCLI Connection Options

Option and Environment Variable	Description
--cacertsfile <certsfile> -t <certs_file> VI_CACERTFILE=<cert_file_path>	ESXCLI commands only. Used to specify the CA (Certificate Authority) certificate file, in PEM format, to verify the identity of the vCenter Server system or ESXi system to run the command on. Can be used, for example, to prevent man-in-the-middle attacks.
--config <cfg_file_full_path> VI_CONFIG=<cfg_file_full_path>	Uses the configuration file at the specified location. Specify a path that is readable from the current directory.
--credstore <credstore>	Name of a credential store file. Defaults to <HOME>/ .vmware/credstore/vicredentials.xml on Linux and <APPDATA>/VMware/credstore/vicredentials.xml on Windows. Commands for setting up the credential store are included in the vSphere SDK for Perl, which is installed with vCLI. The <i>vSphere SDK for Perl Programming Guide</i> explains how to manage the credential store.
--encoding <encoding> VI_ENCODING=<encoding>	Specifies the encoding to be used. Several encodings are supported. <ul style="list-style-type: none"> ■ cp936 (Simplified Chinese) ■ shftjis (Japanese) ■ cp850 (German and French). You can use --encoding to specify the encoding vCLI should map to when it is run on a foreign language system.
--passthroughauth VI_PASSTHROUGHAUTH	If you specify this option, the system uses the Microsoft Windows Security Support Provider Interface (SSPI) for authentication. Trusted users are not prompted for a user name and password. See the Microsoft Web site for a detailed discussion of SSPI. This option is supported only if you are running vCLI on a Windows system and are connecting to a vCenter Server system.
--passthroughauthpackage <package> VI_PASSTHROUGHAUTHPACKAGE=<package>	Use this option with --passthroughauth to specify a domain-level authentication protocol to be used by Windows. By default, SSPI uses the Negotiate protocol, which means that client and server try to negotiate a protocol that both support. If the vCenter Server system to which you are connecting is configured to use a specific protocol, you can specify that protocol using this option. This option is supported only if you are running vCLI on a Windows system and connecting to a vCenter Server system.
--password <passwd> VI_PASSWORD=<passwd>	Uses the specified password (used with --username) to log in to the server. <ul style="list-style-type: none"> ■ If --server specifies a vCenter Server system, the user name and password apply to that server. If you can log in to the vCenter Server system, you need no additional authentication to run commands on the ESXi hosts that server manages. ■ If --server specifies an ESXi host, the user name and password apply to that server. Use the empty string (' ' on Linux and “ ” on Windows) to indicate no password. If you do not specify a user name and password on the command line, the system prompts you and does not echo your input to the screen.
--portnumber <number> VI_PORTNUMBER=<number>	Uses the specified port to connect to the system specified by --server. Default is 443.
--protocol <HTTP HTTPS> VI_PROTOCOL=<HTTP HTTPS>	Uses the specified protocol to connect to the system specified by --server. Default is HTTPS.
--savesessionfile <file> VI_SAVESESSIONFILE=<file>	Saves a session to the specified file. The session expires if it has been unused for 30 minutes.

Table 3-2. vCLI Connection Options (Continued)

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

Table 3-3 lists options not used as connection options that you can use when you run a `vicfg- vCLI` command.

Table 3-3. vCLI Common Options

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

Using vCLI Commands in Scripts

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

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

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

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

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

```
#!/bin/sh
for i in {"esxi_server_a","esx_server_b","esxi_server_c"}
do
    echo "Adding NAS datastore for $i..."
    esxcli --config /host.conf storage nfs add -H <hostname> -s <sharepoint> -v <volumename>
    esxcli --config <config.file> storage nfs list
done
```

ESXCLI Command Hierarchies

This appendix consists of tables of command summaries that illustrate the hierarchy of namespaces and commands (in **bold**) for each ESXCLI namespace. This appendix is not meant to be a reference but facilitates locating the online help for individual commands.

The appendix includes the following topics:

- [“esxcli esxcli Namespace”](#) on page 29
- [“esxcli fcoe Namespace”](#) on page 29
- [“esxcli hardware Namespace”](#) on page 30
- [“esxcli iscsi Namespace”](#) on page 30
- [“esxcli license Namespace”](#) on page 30
- [“esxcli network Namespace”](#) on page 31
- [“esxcli software Namespace”](#) on page 31
- [“esxcli storage Namespace”](#) on page 32
- [“esxcli system Namespace”](#) on page 33
- [“esxcli vm Namespace”](#) on page 33

esxcli esxcli Namespace

The `esxcli esxcli` namespace includes the `esxcli esxcli command list` command.

esxcli fcoe Namespace

`adapter` [**list**]

`nic` [**disable** | **discover** | **list**]

esxcli hardware Namespace

The `esxcli hardware` namespace includes the following namespaces and commands.

cpu [list]	cpuid [get]
	global [get set]
bootdevice [list]	
clock [get set]	
memory [get]	
pci [list]	
platform [get]	

esxcli iscsi Namespace

The `esxcli iscsi` namespace includes the following namespaces and commands.

adapter [get list set]	auth	chap [set get]		
	discovery [rediscover]	sendtarget [add list remove]	auth	chap [get set]
			param [get set]	
		statictarget [add list remove]		
		status get		
	target [list]	portal [list]	auth	chap [get set]
			param [get set]	
	capabilities get			
	firmware [get set]			
	param [get set]			
networkportal [add list remove]	ipconfig [get set]			
physicalnetworkportal [list]	param [get set]			
session [add list remove]	connection list			
ibftboot [get import]				
logicalnetworkportal list				
plugin list				
software [get set]				

esxcli license Namespace

The `esxcli license` namespace includes the `esxcli license oem update` command.

esxcli network Namespace

The `esxcli network` namespace includes the following namespaces and commands.

fence [list]	network [list]	bte [list]		
		port [list]		
firewall [get load refresh set unload]	ruleset [list set]	allowedip[add list remove]		
		rule [list]		
ip [get set]	dns	search [add list remove]		
		server [add list remove]		
	interface [add list remove set]	ipv6 [get set]		
		ipv4 [get set]		
			address [add list remove]	
	connection [list]			
	neighbor [list]			
vswitch	dvs	vmware [list]		
	standard [add list remove set]	policy [get]	failover [set]	
			security [get set]	
			shaping [set]	
		portgroup [add list remove set]	policy [set]	failover [set]
				security [get set]
				shaping [set]
		uplink [add remove]		
nic [down get list set up]				

esxcli software Namespace

The `esxcli software` namespace includes the following namespaces and commands.

sources	profile [get list]
	vib [get list]
acceptance [get set]	
profile [get install update validate]	
vib [get install list remove update]	

esxcli storage Namespace

The `esxcli storage` namespace includes the following namespaces and commands.

core	adapter [list rescan]	stats [get]	
	device [list set setconfig]	vaai [status]	
		detached [list remove]	
		partition [list]	
		stats [get]	
		world [list]	
	path [list set]	stats [get]	
	claiming [reclaim unclaim]		
	claimrule [add convert list load move remove run]		
	plugin [list]		
nmp	psp [list]	fixed	deviceconfig [get set]
		generic	deviceconfig [get set]
			pathconfig [get set]
		roundrobin	deviceconfig [get set]
	satp [list set]	generic	deviceconfig [get set]
			pathconfig [get set]
		rule [add list remove]	
	device [list set]		
	path [list]		
vmfs [upgrade]	snapshot [list mount resignature]	extent [list]	
	extent [list]		
	filesystem [automount list mount rescan unmount]		
	nfs [add list remove]		

esxcli system Namespace

The `esxcli system` namespace includes the following namespaces and commands.

boot	device [get]	
coredump	network [get set]	
	partition [get list set]	
module [get list load set]	parameters [list set]	
process [list]	stats	
		load [get]
		running [get]
secpolicy	domain [list set]	
settings	keyboard	layout [get list set]
	advanced [list set]	
	kernel [list set]	
stats	uptime [get]	
syslog [mark reload]	config [get set]	logger [list set]
visorfs [get]	ramdisk [add list remove]	
	tardisk [list]	
hostname [get set]		
time [get set]		
uuid [get]		
version [get]		
welcomemsg [get set]		

esxcli vm Namespace

The `esxcli vm` namespace includes a `process` namespace with a `list` command and a `kill` command.

Index

A

Active Directory **19**
authentication information **22**

C

command-line connection parameters **24**
configuration files
 for authentication **23**
 usage **23**
connection options **22, 25**
cp936 encoding **26**
creating session files **23**
credential store precedence **22**

D

DCUI **10**
deploying vMA **19**
direct console **10**

E

encoding
 cp936 **26**
 Shift_JIS **26**
ESXCLI command hierarchies **29**
execution options **25**

I

installing vCLI
 Linux **14, 22**
 Windows **17**
installing vMA **19**

L

Linux
 installing vCLI **14, 22**
 running vCLI commands **17, 24**
 vCLI **14**
lockdown mode **25**

M

Microsoft Windows Security Support Provider
 Interface **25**

O

options **25**
order of precedence **22**

P

Perl **13**
precedence **22**

R

required parameters **22**
running commands
 from vMA **19**
 Linux **14, 22**
 Windows **17**

S

scripts with vCLI commands **27**
session files **22, 23**
Shift_JIS encoding **26**
SSPI protocol **25**

U

uninstalling
 Linux **17**
 on Linux **17**
 on Windows **18**
Using **24**
using session files **23**

V

vCLI
 command-line **24**
 configuration files **23**
 environment variables **23**
 execution options **25**
 installing on Linux **14, 22**
 installing on Windows **17**
vCLI package
 installing on Linux **14**
 installing on Windows **18**
 uninstalling **17**
 unpacking **16**
vMA **19**
 environment variables **23**
 installing **19**
 multiple configuration files **24**
vSphere Management Assistant **19**
vSphere SDK for Perl **13**

W

Windows

executing commands **24**
installing vCLI **17**
running vCLI commands **18**
using vCLI **17**