Getting Started with vSphere Command-Line Interfaces

ESXi 6.0

vCenter Server 6.0

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

EN--001469-00

vmware®

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

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

The VMware Web site also provides the latest product updates.

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

docfeedback@vmware.com

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

Contents

About This Book

Getting Started with vSphere Command-Line Interfaces gives an overview of command-line interfaces in vSphere 5.0 and later 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 data center 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/developer/vcli.

- *Command-Line Management in vSphere 5 and vSphere 6 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 host management commands, and explains how to set up software and hardware iSCSI, add virtual switches, place hosts in maintenance mode, and so on. The document includes the same example with the ESXCLI command and with the `vicfg-` command.
- *vSphere Command-Line Interface Reference* is a reference to both ESXCLI commands and `vicfg-` 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.
- *DCLI Reference* is a reference to DCLI commands for managing vCenter services.

The documentation for PowerCLI is available in the vSphere Documentation Center and on the PowerCLI documentation page.

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

The *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 22.

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 a set of ESXi Shell commands, PowerCLI commands, and DCLI commands for management of vCenter services. You can run commands locally, from an administration server, or from scripts.

You can choose the CLI best suited for your needs, and write scripts to automate your management tasks.

This chapter includes the following topics:

Overview of vSphere Command-Line Interfaces

vSphere includes commands for managing different aspects of your environment. The following CLIs are available for managing hosts, either directly or through the vCenter Server system that manages the host.

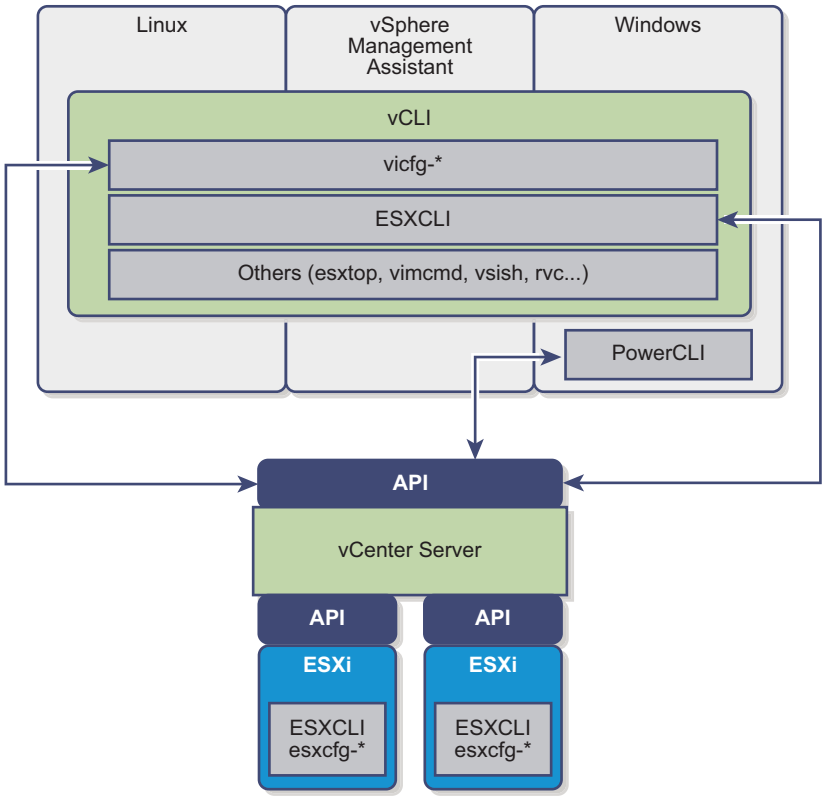

Figure 1-1. vSphere CLIs for host management.

The following command sets are available and are discussed either in this manual or other VMware documentation.

Command set	Description	See
ESXCLI commands	Manage many aspects of an ESXi host. You can run ESXCLI commands remotely or in the ESXi Shell. ■ **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 `–vihost` parameter is required if the host is in lockdown mode. ■ **ESXi Shell**. Run ESXCLI commands in the local ESXi shell to manage that host. You can also run ESXCLI commands from the vSphere PowerCLI prompt by using the `Get–EsxCli` cmdlet.	"Using ESXCLI for Host Management" on page 10 "Installing vCLI" on page 15 *vSphere Command-Line Concepts and Examples* *vSphere Management Assistant Guide* *vSphere Command-Line Interface Reference*
`vicfg–` and other vCLI commands	Allow users to manage hosts remotely. Install the vCLI package on a Windows or Linux system or deploy a vMA virtual machine, and target the ESXi system that you want manipulate. You can run the commands against ESXi systems or against a vCenter Server system. If you target a vCenter Server system, use the `––vihost` option to specify the target ESXi system. **Note:** If the ESXi system is in strict lockdown mode, you must run commands against the vCenter Server system that manages your ESXi system.	*"Installing vCLI" on page 15* *vSphere Command-Line Concepts and Examples* *vSphere Command-Line Interface Reference*

Command set	Description	See
`esxcfg-` commands	Available in the ESXi Shell. `esxcfg-` commands are still included in this release but are deprecated. Migrate to ESXCLI where possible.	*Command-Line Management of vSphere 5 and vSphere 6 for Service Console Users*

DCLI commands	Manage VMware SDDC services.
	DCLI is a CLI client to the vCloud Suite SDK interface for managing VMware SDDC services. A DCLI command talks to a vCloud Suite API endpoint to locate relevant information, and then executes the command and displays result to the user.
	You can run DCLI commands as follows.
	■ **vCenter Server appliance**. Run DCLI commands from the vCenter Server Appliance shell. See "Running DCLI Commands on the vCenter Server Appliance" on page 42.
	■ **vCenter Server Windows command prompt**. Install vCenter Server on a supported Windows system and run DCLI commands from the command prompt.
	■ **vCLI package**.
	  ■ Open a command prompt on a Linux or Windows system on which you installed vCLI. Enter commands into that command prompt, specifying connection options. See "Running DCLI Commands" on page 41.
	  ■ Access the vMA Linux console. DCLI does not support the vi-fastpass connections.
	■ Prepare scripts that include DCLI commands and run the scripts as vCLI scripts from the vCenter Server Windows command prompt or from the vCenter Server Appliance shell.
		"Running DCLI Commands" on page 39
		See the vCloud Suite SDK documentation for information about currently supported services and how they interact.

VMware PowerCLI cmdlets	VMware vSphere PowerCLI provides a Windows PowerShell interface to the vSphere API. vSphere PowerCLI includes PowerShell cmdlets for administering vSphere components. vSphere PowerCLI includes more than 370 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 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 the hoste daemon (`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. **Warning**: If you use a `localcli` command, an inconsistent system state and potential failure can result.	
`pktcap-uw` utility	Enables you to monitor the traffic that flows through physical network adapters, VMkernel adapters, and virtual machine adapters, and to analyze the packet information by using conventional network analysis tools such as Wireshark.	*vSphere Networking* documentation

`dir-cli`
`vecs-cli`
`certool`
	Commands for managing the vCenter Single Sign-On and certificate infrastructure.	*vSphere Security* documentation

`appliancesh`	Enables you to configure and troubleshoot the vCenter Server Appliance and to monitor the processes and services running in the appliance.	*vCenter Server Appliance Configuration* documentation

Using ESXCLI for Host Management

You can manage many aspects of an ESXi host with commands in 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 that are available on a host depends on the host configuration. The *vSphere Command-Line Interface Reference* lists help information for all ESXCLI commands. You can run `esxcli --server <MyESXi> --help` before you run a command on a host to make sure that the command is defined on the host you are targeting.

ESXCLI Syntax

Each ESXCLI 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 Chapter 4, "Running vCLI Host Management Commands," on page 27. Not required when you run the command in the ESXi Shell. If the target server is a vCenter Server system, specify the target ESXi host before any ESXCLI namespaces, commands, and supported options.

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

  IMPORTANT Starting with vSphere 6.0, ESXCLI expects a trust relationship between the target host and the system on which you run the command. You can establish this relationship in one of these ways:

  - Use the `--cacertsfile` option or `VI_CACERTFILE` variable
  - Store the thumbprint in the session file.
  - Specify the thumbprint with the `--thumbprint` option or `VI_THUMBPRINT` variable.

  You can pass in the thumbprint that is returned in the error if you trust the host you are targeting. See "Trust Relationship Requirement for ESXCLI Commands" on page 33 for an example.

- **namespace**. Groups ESXCLI commands. vSphere 5.0 and later support nested namespaces.
- **command**. Reports on or modifies state on the system.

Examples

```
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
```

- **options**. Many commands support one or more options, displayed in the help or the vCLI Reference. For some commands, multiple option values, separated by spaces, are possible.

Example

```
esxcli system module parameters set -m <module> -p "a=1 b=1 c=1"
```

Running ESXCLI vCLI Commands

You can run an ESXCLI vCLI command in the ESXi Shell for troubleshooting and 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. Specify connection options to run commands against an ESXi host directly, or target a vCenter Server system and specify the ESXi host to run the command against. See "Installing vCLI" on page 15.

  NOTE Starting with vSphere 6.0, a trust relationship must exist between the host from which you run ESXCLI commands and the target ESXi host or vCenter Server system. See Appendix 4, "Trust Relationship Requirement for ESXCLI Commands," on page 33.

See Chapter 4, "Running vCLI Host Management Commands," on page 27.

ESXCLI Command Support when Host and vCLI Version Do Not Match

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. For example:

- 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. ESXCLI 5.1 commands that were included in ESXCLI 5.0 are also supported.
- If you run commands against ESXi 5.1 hosts, ESXCLI 5.1 and ESXCLI 5.0 commands are supported.

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

Using PowerCLI to Manage Hosts and Virtual Machines

VMware vSphere PowerCLI contains snap-ins and modules based on Microsoft PowerShell for automating vSphere and vCloud Director administration. PowerCLI provides C# and PowerShell interfaces to vSphere and other VMware product administration.

vSphere PowerCLI is based on Microsoft PowerShell and uses the PowerShell basic syntax and concepts. Microsoft PowerShell is both a command-line and scripting environment, designed for Windows. It uses the .NET object model and provides administrators with system administration and automation capabilities. To work with PowerShell, you run commands, which are called cmdlets.

PowerShell supports features such as pipelines, wildcards, and easy access to command-line help.

You can use ESXCLI commands from the vSphere PowerCLI console, as follows:

- Through the cmdlet, which provides direct access to the ESXCLI namespaces, applications, and commands.
- Through .NET methods, which you use to create managed objects that correspond to specific ESXCLI applications. To access the ESXCLI, you can call methods on these managed objects.

  NOTE To run an ESXCLI command from PowerCLI, you must provide values for all parameters. If you want to omit a given parameter, pass $null as its argument.

See the *vSphere PowerCLI User's Guide* in the vSphere documentation center.

Using DCLI to Manage vCenter Services

The DCLI command set allows you to manage vCenter services that are new in vSphere 6.0. You cannot manage services that were part of vSphere 5.5 from DCLI. DCLI is not a host management CLI.

DCLI (Datacenter CLI) is a CLI client of the vCloud Suite SDK. DCLI works like this:

1 The user runs a command in the DCLI directory and specifies a user name.

2 If the user is not yet authenticated, DCLI prompts for a password.

3 The user specifies a password.

4 The command connects to the vCenter Single Sign-On service and checks whether the user specified on the command-line or in a certificate store file can authenticate.

5 If the user can authenticate, DCLI communicates with the vCenter Server and execute the vCloud Suite SDK command that corresponds to the DCLI command. Different vCenter Server systems support different services.

6 DCLI displays the result or an error to the user.

You can run DCLI commands as follows.

- **vCLI package**. Install the vCLI package on the server of your choice, or deploy a vMA virtual machine. You can then run DCLI commands against an endpoint. See "Running DCLI Commands" on page 41.
- **vCenter Server appliance**. Run DCLI commands from the vCenter Server Appliance shell. See "Running DCLI Commands on the vCenter Server Appliance" on page 42.
- **vCenter Server Windows command prompt**. Install vCenter Server on a supported Windows system and run DCLI commands from the command prompt.

DCLI Syntax

Each DCLI command uses the same syntax.

The command name can be followed by connection and formatting options, each preceded by a + sign. You also specify the name space, the command, and the command options. Name spaces are nested.

NOTE The order in which DCLI options are provided on the command line is not important. However, you must specify DCLI options with a plus (+) and command-specific options with a minus (-).

```
dcli +[DCLI options] <namespace> [<namespace> ...] <cmd> --[cmd option] [option value]
```

- **DCLI options**. Predefined options for connection information including the vCloud Suite SDK endpoint and formatting options. Always preceded by a + sign.

  Not required when you run the command in the vCenter Server Appliance shell or from the command prompt of a vCenter Server Windows installation.
- **namespace**. Groups DCLI commands. Namespaces correspond to the vCloud Suite SDK namespaces and are nested.
- **command**. Reports on or modifies state on the system.
- **option and value**. Command option and value pairs.

Example

```
$dcli +server <vcenter-IP> com vmware cis tagging tag list
```

vCLI Package Contents

vCLI is not a command set but a package of several command sets. You usually install vCLI on an administration server and run scripts from there against other hosts or, for DCLI, against vCenter Server systems. Some vCLI commands can also be run locally on the ESXi host or the vCenter Server system.

When you install the vCLI package, the following command sets become available.

- **DCLI Commands.** The DCLI commands are new in vSphere 6.0, and are available for managing vCenter services that are new in vSphere 6.0. These commands are available as part of vCLI, from the vCenter Server Virtual Appliance, and from the command-prompt of a vCenter Server Windows installation.
- **Host Management Commands.** Includes the following command sets.

- **ESXCLI commands.** The ESXCLI commands included in the vCLI package are equivalent to the ESXCLI commands available in the ESXi Shell.
- **vicfg- commands.** The `vicfg–` command set is similar to the deprecated `esxcfg–` command set in the ESXi Shell.
- **Miscellaneous commands**. A small set of commands for managing and monitoring ESXi hosts, including `vmkfstools` and `resxtop`. In many cases, equivalent but slightly different commands are available 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 15.
- 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 host management commands in the set against ESXi hosts. You can run the DCLI commands against a server by specifying the IP address and can manage the services associated with that server.

You must specify connection parameters when you run a vCLI command. The connection parameters differ for DCLI commands and for other commands. See "Running vCLI Host Management Commands" on page 27 and "Running DCLI Commands" on page 41.

Installing vCLI 2

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

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

- **Installable Package.** Install a vCLI package on a physical or virtual machine. See "Installing the vCLI Package on Red Hat Enterprise Linux" on page 18, "Installing vCLI on Linux Systems with Internet Access" on page 19, and "Installing and Uninstalling vCLI on Windows" on page 21.

  The vCLI installer installs both vSphere SDK for Perl and vCLI because many vCLI commands run on top of the vSphere SDK for Perl. The contents of the installer package differs for different platforms.

Platform	Installation Process
Windows	The installation package includes vCLI, vSphere SDK for Perl, and prerequisite Perl modules.
Red Hat Enterprise Linux	You must install required software. See "Installing Required Prerequisite Software for Red Hat Enterprise Linux" on page 18. The installer for RHEL prompts you whether you want to install other missing modules from the Internet or from the package. ■ If you have Internet access, you can have the installer download Perl modules from CPAN. ■ The installer can instead install Perl modules that it does not find on your system from the installer package.
SLES and Ubuntu	You must install required software and you must have Internet access. See "Installing Required Prerequisite Software for Linux Systems with Internet Access" on page 19. The installer downloads other Perl modules from CPAN.

After installation, 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. You can also write scripts and manage your vSphere environment using those scripts.

- **vSphere Management Assistant (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 host management commands 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 22.

Overview of Linux Installation Process

The installation script for vCLI is supported on the Linux distributions that are listed in the *Release Notes*.

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 (`libssl-devel` package) is not included in the default Linux distribution. See "Installing Required Prerequisite Software for Red Hat Enterprise Linux" on page 18 and "Installing Required Prerequisite Software for Linux Systems with Internet Access" on page 19.
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 `libxml2` package is not included in the default Linux distribution. See "Installing Required Prerequisite Software for Red Hat Enterprise Linux" on page 18 and "Installing Required Prerequisite Software for Linux Systems with Internet Access" on page 19.
uuid	Included in `uuid-devel` for SLES 11 and in `e2fsprogs-devel` 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 Red Hat Enterprise Linux" on page 18 and "Installing Required Prerequisite Software for Linux Systems with Internet Access" on page 19 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-2.037
- Compress-Zlib-2.037
- IO-Compress-Zlib-2.037
- Compress-Raw-Zlib-2.037
- Archive-Zip-1.28
- 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-0.03
- SOAP-Lite-0.710.08
- HTML-Parser-3.60
- version-0.78
- Class-MethodMaker-2.10
- JSON-PP-2.27203
- Devel-StackTrace-131
- Class-Data-Inheritable-0.08
- Convert-ASN1-0.26
- Cyrpt-OpenSSL-RSA-0.28
- Crypt-X509-0.51
- Exception-Class-1.37
- MIME-Base64-3.14
- UUID-Random-0.04
- Socket6-023
- IO-Socket-INET6-2.71
- Net-INET6Glue-0.600_1

Earlier versions of libwww-perl include the LWP-Protocol-https module. More recent versions of libwww-perl do not include the LWP-Protocol-https module and you have to install that module.

NOTE If you intend to run vCLI commands with SSL certification, be sure to check that LWP::UserAgent 6.00 or later is installed. The installer does not check this module, and earlier versions do not work with SSL.

4 The installer proceeds depending on the Linux distribution.

Linux distribution	Installer behavior
RHEL (No Internet access)	On RHEL, the installer allows you to install Perl modules with CPAN if Internet access is available. If no Internet access is available, and a module is not currently on your system, the installer installs it. If a different version of a module is found, the installer does not install it and proceeds with installation. At the end of the installation process, the installer informs you if the version on the system does not match the recommended version, and recommends that you install the version that vCLI was tested with. You can install the modules using the package installer for your platform, the installation CD, or CPAN. **Note:** The installer does not overwrite existing versions of recommended Perl modules. You must explicitly update those modules yourself.
All Linux distributions (Internet access)	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 "Installing vCLI on Linux Systems with Internet Access" on page 19. ■ 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. **Note:** 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 Red Hat Enterprise Linux" on page 18 and "Installing the vCLI Package on a Linux System with Internet Access" on page 20.

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 the vCLI Package on Red Hat Enterprise Linux

vCLI is supported on Red Hat Enterprise Linux versions that are listed in the *Release Notes*. On RHEL, the vSphere SDK for Perl installer prompts you whether you want to install required Perl modules from the installation package or from CPAN. Follow these steps to install the software.

1 Install required prerequisite software. See "Installing Required Prerequisite Software for Red Hat Enterprise Linux" on page 18.

2 When prompted, direct the installer to install additional prerequisites from the installation package (see "Installing the vCLI Package on RHEL (No Internet Access)" on page 18) or from CPAN (see "Installing the vCLI Package on a Linux System with Internet Access" on page 20).

Installing Required Prerequisite Software for Red Hat Enterprise Linux

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

Required Software

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

RHEL 6.3 32 bit
```
yum install e2fsprogs-devel libuuid-devel
yum install perl-XML-LibXML
```

RHEL 6.3 64 bit
```
yum install e2fsprogs-devel libuuid-devel
yum install glibc.i686
yum install perl-XML-LibXML
```

Recommended Perl Modules

When the installer finishes, it might issue a warning that the version of a module installed on your system does not match the version with which vCLI was tested. Install that version using `yum` or CPAN to resolve the issue. See "Overview of Linux Installation Process" on page 16 for a complete list of modules.

NOTE The installer does not overwrite existing Perl modules.

Installing the vCLI Package on RHEL (No Internet Access)

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

To remove previous versions of vCLI

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

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

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

To install vCLI on RHEL

1 Untar the vCLI binary that you downloaded.

```
tar -zxvf VMware-vSphere-CLI-6.X.X-XXXXX.XXXX.x86_64.tar.gz
```

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

2 Log in as superuser and run the installer:

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

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

4 To install Perl modules locally, type **yes** and press Enter.

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

A complete installation process has the following result:

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

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

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

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

After you install the vCLI, you can test the installation by running a command from the command prompt. See "Running Host Management Commands from a Linux System" on page 37.

Installing vCLI on Linux Systems with Internet Access

Before you can install the vCLI package on a Linux system with Internet access, that system must meet 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
```

Installing Required Prerequisite Software for Linux Systems with Internet Access

If required prerequisite software is not installed, the installer stops and requests that you install it. Installation of prerequisite software depends on the platform that you are using. See the *Release Notes* for the supported versions of each Linux platform.

Table 2-1. Installing Required Prerequisite Software

Platform	Installation

RHEL 6.3 32 bit	Find the required modules on the installation DVD, or use yum to install them.
	`yum install e2fsprogs-devel libuuid-devel`
	`yum install perl-XML-LibXML`

RHEL 6.3 64 bit	Find the required modules on the installation DVD, or use yum to install them.
	`yum install e2fsprogs-devel libuuid-devel`
	`yum install glibc.i686`
	`yum install perl-XML-LibXML`

SUSE Enterprise	Install the prerequisite packages from the SLES 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. ■ **SLES 11 64 bit.** `yast -i openssl-devel libuuid-devel libuuid-devel-32bit` ■ **SLES 11 32 bit**. `yast -i openssl-devel libuuid-devel` Some users might be authorized to use the Novell Customer Center and use `yast` to retrieve missing packages from there.
Ubuntu	1. Connect to the Internet. 2. Update the local repository of libraries from a terminal window. `sudo apt-get update` 3. Install the required libraries from a terminal window. ■ 32 bit. `sudo apt-get install build-essential gcc uuid uuid-dev perl libssl-dev perl-doc liburi-perl libxml-libxml-perl libcrypt-ssleay-perl` ■ 64 bit. `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` For Ubuntu 10.04 64 bit, the `resxtop` and ESXCLI commands do not work if you do not install the 32-bit compatibility libraries.

Installing the vCLI Package on a Linux System with Internet Access

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

To install vCLI

1 Log in as root.

2 Untar the vCLI binary that you downloaded.

```
tar -zxvf VMware-vSphere-CLI-6.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. Establishing a connection might take a long time.

6 On RHEL, when prompted to install precompiled Perl modules, type **`no`** and press Enter to use CPAN

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

7 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.

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 Windows platforms that are listed in the *Release Notes*.

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 `vcredist_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 in the **Automation Tools and SDKs** section of the **Drivers & Tools** tab of the vSphere download 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.

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*.

Running Host Management Commands in the ESXi Shell 3

In most cases, installing vCLI and running commands from a remote system, with one or more hosts as targets, is recommended. However, for maintenance and troubleshooting tasks you might prefer to run ESXCLI commands in the ESXi Shell or connect to the ESXi Shell with SSH.

You first establish access, and can then run commands.

ESXi Shell Access with the Direct Console

An ESXi system includes a Direct Console User Interface (DCUI) that allows you to start and stop the system and to perform a limited set of maintenance and troubleshooting tasks. The direct console allows access to the ESXi Shell, which is disabled by default. You can enable the ESXi Shell in the direct console or by using the vSphere Web 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 23.
- 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. See "Remote ESXi Shell Access with SSH" on page 24.

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 in the ESXi Shell directly or through SSH only in troubleshooting situations.

Enabling Local ESXi Shell Access

You can enable the ESXi Shell from the direct console or from the vSphere Web Client or 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 Interface, you can enable the ESXi Shell from the vSphere Web Client.

To enable the ESXi Shell from the vSphere Web Client or the vSphere Client

1 Select the host, click **Manage**, and keep Settings selected.

2 Click **Security Profile**.

3 In the Services section, click **Edit.**

4 Select **ESXi Shell**.

   - To temporarily start or stop the service, click the **Start** or **Stop** button.
   - To change the Startup policy across reboots, select **Start and stop with host** and reboot the host.

5 Click **OK**.

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

ESXi Shell Timeout

The ESXi Shell supports a timeout for ESXi Shell availability and a timeout for idle ESXi Shell sessions.

- **Availability timeout**: The availability timeout setting is the amount of time that can elapse before you must log in after the ESXi Shell is enabled. After the timeout period, the service is disabled and users are not allowed to log in.
- **Idle timeout**: If a user enables the ESXi Shell on a host, but forgets to log out of the session, the idle session remains connected indefinitely

You can set both timeout values from the Direct Console User Interface, from the vSphere Web Client, or from the vSphere Client. See the *vSphere Security* document in the vSphere Documentation Center for detailed instructions.

Using the ESXi Shell

After you enable the ESXi Shell in the direct console, you can use it from the 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 SSH connections are enabled for your ESXi host, 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 Web 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**.

To enable SSH from the vSphere Web Client

1 Select the host, click **Manage**, and keep Settings selected.

2 Click **Security Profile**.

3 In the Services section, click **Edit.**

4 Select **SSH**.

- To temporarily start or stop the service, click the **Start** or **Stop** button.
- To change the Startup policy across reboots, select **Start and stop with host** and reboot the host.

5 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.

Lockdown Mode

To increase the security of your ESXi hosts, you can put them in lockdown mode. In lockdown mode, all operations must be performed through vCenter Server. By default, only the vCenter Server system, represented by the vpxuser user, has authentication permissions. No other users can perform operations against a host in Lockdown Mode.

vSphere 5.x and later supports normal lockdown mode, as discussed in the vSphere 5.x documentation center. vSphere 6.0 and later supports finer grained management:

- In normal lockdown mode, you can add users to the DCUI.Access advanced option which can access the Direct Console User Interface regardless of their privileges on the host. Starting with vSphere 6.0, you can also use the vSphere Web Client to add Exception users, which can access the Direct Console User Interface if they have host management privileges.
- In strict lockdown mode, users cannot access the Direct Console User Interface. If vCenter Server becomes unavailable, the host can no longer be managed.

When a host is in normal or strict lockdown mode, you cannot run vSphere CLI commands against the host directly. Instead, you target the vCenter Server system that manages the host with the `--server` option and specify the ESXi host with the `--vihost` option.

When you enable strict lockdown mode, the Direct Console User Interface service is disabled.

You can enable lockdown mode using the Add Host wizard to add a host to vCenter Server, using the vSphere Web Client to manage a host, or using the Direct Console User Interface (DCUI).

See the *vSphere Security* documentation for details on Lockdown Mode in vSphere 6.0.

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 23.

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.

Running vCLI Host Management Commands

4

You can run vSphere Command-Line Interface (vCLI) host management commands from the command line of the system where you installed the package, from the vMA command line, and from scripts.

Host management commands, which include ESXCLI and `vicfg-` commands, require at a minimum the target server to run the command on. Users must authenticate to the host, and can only perform tasks that they are authorized to perform.

NOTE See "Running DCLI Commands" on page 39 for information about DCLI commands, which you can use to manage vCenter Server services.

This chapter includes the following topics:

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 `--vihost` option to specify the ESXi target. See "vCLI and Lockdown Mode" on page 32.

Overview of Running vCLI Host Management Commands

You can run vCLI commands interactively or in scripts, and you can target the host directly or target a vCenter Server system that manages the host.

Targeting the Host Directly

You can target the host directly from an administration server on which you installed VCLI, use vMA, or run scripts.

- Open a command prompt on a Linux or Windows system on which you installed vCLI. Enter commands into that command prompt, specifying connection options. See "Authenticating Directly to the Host" on page 30.

- 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 system that has the vCLI package installed or from the vMA Linux console. See "Using vCLI Commands in Scripts" on page 36.

NOTE Different command sets in the vCLI package require different connection options.

When you run commands against an ESXi host, you must be authenticated for that host.

Target a Host That is Managed by a vCenter Server System

When you target a host that is managed by a vCenter Server system, you can run commands in different ways.

- Specify the vCenter Single Sign-On service with `--psc` and, if multiple vCenter Server systems are associated with the vCenter Single Sign-On service, the vCenter Server system with `--server`. Specify also the host with `--vihost`.
- Specify the vCenter Server system with `--server` and the ESXi host with `--vihost.`
- Specify only the ESXi host with `--vihost`.

When you can authenticate to a vCenter Single Sign-On service or to a vCenter Server system, you can target all ESXi hosts that vCenter Server manages without additional authentication. See "Authenticating Through vCenter Server and vCenter Single Sign-On" on page 29.

Protecting Passwords

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 host management 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` option. See "Using a Session File" on page 30.
- Target a vCenter Server system and authenticate to vCenter Single Sign-On. You can save the corresponding session and use it for subsequent connections. See "Authenticating Through vCenter Server and vCenter Single Sign-On" on page 29.
- Use variables or configuration files.
- 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 a known Active Directory user, 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*.

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

IMPORTANT Administrators can place ESXi hosts in lockdown mode for enhanced security. By default, not even the root user can run vCLI commands directly against ESXi hosts in lockdown mode. See "vCLI and Lockdown Mode" on page 32 and the *vSphere Security* documentation.

Order of Precedence for vCLI Host Management Commands

When you run a vCLI host management command, authentication happens in the order of precedence shown in Table 4-1. This order of precedence always applies. That means, for example, that you cannot override an environment variable setting in a configuration file.

NOTE Available options and order of precedence are different for DCLI. See "Order of Precedence for DCLI Authentication" on page 43.

If you are authenticating through vCenter Single Sign-On, the order of precedence is preserved, for example, information you specify on the command line overrides information in an environment variable.

Table 4-1. vCLI Authentication Precedence

Authentication	Description	See
Command line	Password (`--password`), session file (`--sessionfile`), or configuration file (`--config`) specified on the command line.	"Using a Session File" on page 30
Environment variable	Password specified in an environment variable.	"Using Environment Variables" on page 30
Configuration file	Password specified in a configuration file.	"Using a Configuration File" on page 31
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 32
Credential store	Password retrieved from the credential store.	*vSphere Web Services SDK Programming Guide* and *vSphere SDK for Perl Programming Guide*.
Prompt the user for a password.	Password is not echoed to screen.	

Authenticating Through vCenter Server and vCenter Single Sign-On

For all ESXi hosts that are managed by a vCenter Server system that is integrated with vCenter Single Sign-On 6.0 and later, you can authenticate directly to the vCenter Server system, or you can authorize to vCenter Server through vCenter Single Sign-On.

Best practice is to authenticate through vCenter Single Sign-On. The vCenter Single Sign-On service is included in the Platform Services Controller. The Platform Services Controller can be embedded in your vCenter Server installation, or one Platform Services Controller can handle authentication, certificate management, and some other tasks for multiple vCenter Server systems.

NOTE You cannot use this approach if vCenter Server is integrated with vCenter Single Sign-On 5.0.

You use the `--psc` option and, optionally, the `--server` option.

- `psc` - Specifies the Platform Services Controller instance associated with the vCenter Server system that manages the host.
- `server` - Specifies the vCenter Server system that manages the host. Required if the Platform Services Controller instance is associated with more than one vCenter Server system.
- `vihost` - Specifies the ESXi host, as in earlier versions of vCLI.

Examples

```
vicfg-nics -l --username <sso_username> --password "<admin_pwd>" --server <vc_HOSTNAME_OR_IP>
              --psc <psc_HOSTNAME_OR_IP> --vihost <esxi_HOSTNAME_OR_IP>
esxcli --server <vc_HOSTNAME_OR_IP> --vihost <esxi_HOSTNAME_OR_IP> --username <USERNAME>
              --password <PASSWORD> --psc <psc_HOSTNAME_OR_IP> hardware clock get
```

If the specified user is known to vCenter Single Sign-On, a session is created. You can save the session with the `--savesessionfile` argument, and later use that session with the `--sessionfile` argument. For example, you can save the session by running this command:

```
vicfg-nics -l --username <sso_username> --password "<admin_pwd>" --server <vc_HOSTNAME_OR_IP>
              --psc <psc_HOSTNAME_OR_IP> --vihost <esxi_HOSTNAME_OR_IP>
```

Using a session file results in less overhead and better performance than connecting to the Platform Services Controller repeatedly.

Authenticating Directly to the Host

vCLI offers several options for authenticating directly to the host.

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-vcli/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 36 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 4-2, "vCLI Connection Options," on page 34.

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_PSC = XX.XXX.XXX.XX
VI_USERNAME = administrator@vsphere.local
VI_PASSWORD = admin_password
VI_PROTOCOL = https
VI_SERVER = my_vc
```

If you have set up your system to run this file, you can run scripts against the specified ESXi host 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_HOSTNAME_OR_IP> --username <privileged_user> --password <pw> --vihost
               <esxi_HOSTNAME_OR_IP> <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_HOSTNAME_OR_IP> --username <privileged_user> --password <pw>
               --vihost <esxi_HOSTNAME_OR_IP> --<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 <esxi_HOSTNAME_OR_IP> --username snow\-white --password dwarf\$ network ip
               interface list
esxcli --server <esxi_HOSTNAME_OR_IP> --username snow\-white --password 'dwarf$' network ip
               interface list
vicfg-mpath --server <esxi_HOSTNAME_OR_IP> --username snow\-white --password dwarf\$ --list
vicfg-mpath --server <esxi_HOSTNAME_OR_IP> --username 'snow-white' --password 'dwarf$' --list
```

Windows

```
esxcli --server <esxi_HOSTNAME_OR_IP> --username "snow-white" --password "dwarf$" network ip
               interface list
vicfg-mpath.pl --server <esxo_HOSTNAME_OR_IP> --username "snow-white" --password "dwarf$" --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. 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_HOSTNAME_OR_IP> --passthroughauth --passthroughauthpackage "Kerberos"
    --vihost <esxi_HOSTNAME_OR_IP> network ip interface list

vicfg-mpath.pl --server <vc_HOSTNAME_OR_IP> --passthroughauth --passthroughauthpackage
               "Kerberos" --vihost <esxi_HOSTNAME_OR_IP> --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 can disable 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 Web 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:

- `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.

Trust Relationship Requirement for ESXCLI Commands

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

To establish the trust relationship, you have these options.

Downloading and Installing the vCenter Server Certificate

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

To download the certificate

1 Type the URL of the vCenter Server system or vCenter Server Virtual Appliance into a Web Browser.

2 Click the **Download trusted root certificates** link.

3 Change the extension of the downloaded file to .zip. (The file is a ZIP file of all certificates in the TRUSTED_ROOTS store).

4 Extract the ZIP file.

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

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

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

Using the --cacertsfile Option

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

Using the --thumbprint Option

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

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

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

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

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

Using the Credential Store

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

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

To establish trust for a user with the credential store

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

```
credstore_admin.pl add --server <esxi_HOSTNAME_OR_IP> --username <user> --password <pwd>
```

2 Add the thumbprint for the target ESXi host. This thumbprint was returned in the error when you attempted to connect to the host.

```
credstore_admin.pl add --server <esxi_HOSTNAME_OR_IP> --thumbprint <thumbprint>
```

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

Common Options for vCLI Host Management Command Execution

Table 4-2 lists options that are available for all vCLI host management commands in alphabetical order. The table includes options for use on the command line and variables for use in configuration files. Options for executing DCLI commands are different.

IMPORTANT Starting with vSphere 5.5, vCLI supports both IPv4 and IPv6 connections.

See "Running Host Management Commands from a Windows System" on page 37 and "Running Host Management Commands from a Linux System" on page 37.

Table 4-2. 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.
	In vCLI 6.0 and later, you can only run ESXCLI commands if a trust relationship exists between the host you are running the command on and the system you are targeting with the `--server` option (ESXi host or vCenter Server system). You can establish the trust relationship by specifying the CA certificate file or by passing in the thumbprint for each target server (ESXi host or vCenter Server system).

`--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>` `VI_CREDSTORE=<credstore>`	Name of a credential store file. Defaults to `<HOME>/.vmware/credstore/vicredentials.xml` on Linux and `<APPDATA>/VMware/credstore/vicredentials.xml` on Windows. Commands for setting up the credential store are included in the vSphere SDK for Perl, which is installed with vCLI. The *vSphere SDK for Perl Programming Guide* explains how to manage the credential store.
`--encoding <encoding>` `VI_ENCODING=<encoding>`	Specifies the encoding to be used. Several encodings are supported. ■ `utf8` ■ `cp936` (Simplified Chinese) ■ `shftjis` (Japanese) ■ `iso-885901` (German). You can use `--encoding` to specify the encoding vCLI should map to when it is run on a foreign language system.

Table 4-2. vCLI Connection Options (Continued)

Option and Environment Variable	Description
`--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 connecting to a vCenter Server system.
`--passthroughauthpackage <package>` `VI_PASSTHROUGHAUTHPACKAGE=<package>`	Use this option with `--passthroughauth` to specify a domain-level authentication protocol to be used by Windows. By default, SSPI uses the `Negotiate` protocol, which means that client and server try to negotiate a protocol that both support. If the vCenter Server system to which you are connecting is configured to use a specific protocol, you can specify that protocol using this option. This option is supported only if you are running vCLI on a Windows system and connecting to a vCenter Server system.
`--password <passwd>` `VI_PASSWORD=<passwd>`	Uses the specified password (used with `--username`) to log in to the server. ■ If --server specifies a vCenter Server system, the user name and password apply to that server. If you can log in to the vCenter Server system, you need no additional authentication to run commands on the 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.
`--psc <hostname_or_IP>` `VI_PSC=<hostname_or_IP>`	Host name or IP address of the Platform Services Controller instance that is associated with the vCenter Server system that manages the host. In many cases, the Platform Services Controller is embedded in the vCenter Server system, but external Platform Services Controller instances are supported as well. For those cases, use the `--server` option to specify the vCenter Server system that manages the host. This option implies user authentication with vCenter Single Sign-On. The user you specify must be able to authenticate to vCenter Single Sign-On.
`--savesessionfile <file>` `VI_SAVESESSIONFILE=<file>`	Saves a session to the specified file. The session expires if it has been unused for 30 minutes.
`--server <server>` `VI_SERVER=<server>`	Uses the specified ESXi or vCenter Server system. Default is `localhost`. If `--server` points to a vCenter Server system, you can also specify the `--psc` option to log in to the vCenter Server system with vCenter Single Sign-On. Use the `--vihost` option to specify the ESXi host that you want to run the command against. See "Authenticating Through vCenter Server and vCenter Single Sign-On" on page 29.
`--servicepath <path>` `VI_SERVICEPATH=<path>`	Uses the specified service path to connect to the ESXi host. Default is `/sdk/webService.`
`--sessionfile <file>` `VI_SESSIONFILE=<file>`	Uses the specified session file to load a previously saved session. The session must be unexpired.
`--thumbprint <thumbprint>` `VI_THUMBPRINT=<thumbprint>`	Expected SHA-1 host certificate thumbprint if no CA certificates file is provided in the `--cacertsfile` argument. The thumbprint is returned by the server in the error message if you attempt to run a command without specifying a thumbprint or certificate file.
`--url <url>` `VI_URL=<url>`	Connects to the specified vSphere Web Services SDK URL.

Table 4-2. vCLI Connection Options (Continued)

Option and Environment Variable	Description
`--username <u_name>` `VI_USERNAME=<u_name>`	Uses the specified user name. ■ If `--server` specifies a vCenter Server system, the user name and password apply to that server. If you can log in to the vCenter Server system, you need no additional authentication to run commands on the ESXi hosts that server manages. ■ If `--server` 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.
`--vihost <host>` `-h <host>`	When you run a vCLI command with the `--server` option pointing to a vCenter Server system, use `--vihost` 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 `<cmd> --help`.

Table 4-3 lists options not used as connection options that you can use when you run a `vicfg-` vCLI command.

Table 4-3. vCLI Common Options

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

Using 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 data store becomes available in your environment, you must make that data store available to each ESXi host. The following sample script illustrates how to make a NAS data store 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 adds the NAS data store to each host defined in VIHOSTS.

```
#!/bin/bash

VI_CONFIG_FILE=/home/admin/viconfig
VIHOSTS=(esxi_server_a esx_server_b esxi_server_c)


for VIHOST in ${VIHOSTS[@]}
do
  echo "Adding NAS datastore for ${VIHOST} ..."
  esxcli --config ${VI_CONFIG_FILE} storage nfs add --host ${VIHOST} --share <share point>
              --volume-name <volume name>
  esxcli --config ${VI_CONFIG_FILE} storage nfs list
done
```

Running Host Management Commands from a Windows System

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:

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

The system prompts you for a user name and password.

Running Host Management Commands from a Linux System

After installation, you can run vCLI commands and vSphere SDK for Perl utility applications at 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:

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

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

Running DCLI Commands 5

You can run DCLI commands as vCLI commands, from the vCenter Server Appliance shell, and from the command prompt of a vCenter Server Windows installation.

IMPORTANT Authentication options for DCLI commands differ from options for vCLI host management commands.

IMPORTANT Users who run DCLI commands to monitor and manage vCenter services must have the appropriate privileges.

- When you run DCLI commands included with vCLI, you must be a user who can authenticate to vCenter Single Sign-On and who is also authorized to perform the service, for example, manage vCenter tags.
- When you run DCLI commands from the vCenter Server Appliance shell, DCLI allows you to run many commands without additional authentication. However, for management of certain services, you might be prompted for a user name and password.

Overview of Running DCLI Commands

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

- Run DCLI commands locally from the vCenter Server Appliance shell.
- Run DCLI commands locally from your vCenter Server on Windows command prompt.
- Run DCLI commands that are included in the vCLI package.
  - 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. DCLI does not support the vi-fastpass connections available from vMA.
- Prepare scripts that contain DCLI commands. Then run the scripts as vCLI scripts, from the vCenter Server Windows command prompt, or from the vCenter Server Appliance shell. Use the credential store options to authenticate, passwords are not supported in scripts.

DCLI Syntax

Each DCLI command uses the same syntax.

The command name is followed by DCLI connection and formatting options, each preceded by a + sign. After the DCLI options come the name space, the command, and the command options, as in the following example:

```
dcli +[DCLI options] <namespace> [<namespace> ...] <cmd> --[cmd option] [option value]
```

- **DCLI options**. Predefined options for connection information and formatting options. Always preceded by a + sign.

  Not required when you run the command on the local host at the Windows command prompt or the vCenter Server Appliance shell.
- **namespace**. Groups DCLI commands. Namespaces correspond to the vCloud Suite SDK name spaces.
- **command**. Reports on or modifies state on the system.
- **option and value**. Command option and value pairs preceded by minus minus (--).

Example

```
$dcli +server my_remote_vc +username user42 com vmware cis tagging tag list
```

DCLI Options

You can run each DCLI command with connection or formatting options preceded by a +.

For many of the options, you can instead use variables, discussed in Table 5-3, "Variables Supported by DCLI," on page 43.

```
dcli [+server SERVER_IP]
    [+interactive]
    [+prompt PROMPT]
    [+ssl-cert-file SSL_CERT_FILE]
    [+ssl-key-file SSL_KEY_FILE]
    [+cacert-file CACERT_FILE]
    [+more]
    [+formatter {simple,table,xml,json,html,csv}]
    [+loglevel {debug, info, warning, error}]
    [+username USERNAME] [+password]
    [+credstore-file CREDSTORE_FILE]
    [+credstore-add | +credstore-remove | +credstore-list]
    [+session-manager SESSION_MANAGER] [args [args ...]]
```

These options allow you to provide the following information. If you are entering options interactively, tab completion is supported on Linux systems. In all cases, you can specify a partial option as long as the option is not ambiguous. For example, `+i` indicates interactive, but you have to specify, at a minimum, `+credstore-a` to disambiguate that option.

Table 5-1. DCLI Options

Option	Description	Default
`server`	The vCenter Server system to which DCLI connects.	`localhost`
`interactive`	Runs DCLI in interactive shell mode, which supports tab completion of commands, options, and some option values. It also supports saving the command history across DCLI sessions. Interactive mode is faster because DCLI caches the list of commands available on a vCenter Server system.	
`prompt`	Prompt that the interactive shell uses.	dcli>

`ssl-key-file`
`ssl-cert-file`
`cacert-file`
	If the CLI client connects to a vCenter Server system that is using HTTPS connections, you can use these options to provide SSL authentication options.	

`more`	Displays page-wise output.	

Table 5-1. DCLI Options

Option	Description	Default
`formatter`	Output formatter, one of the following: ■ simple ■ table ■ xml ■ json ■ html ■ csv	Default is table for lists of structures and simple for all other output.
`loglevel`	The log level, one of debug, info, warning, or error.	info
`username`	If you run from the local shell, most DCLI commands do not require the user name. If you are running vCLI commands, the user you specify must be able to authenticate to the vCenter Server system. The user you specify must have the privileges to perform the task, as specified through vCenter Server roles. You are prompted for the password. The password is not echoed to screen.	
`credstore-file`	Path to the credential store file to use for credential store operations or for reading login credentials. Use this option only if the default credential store file name does not work in your environment. By default, the credential store file is in the `.dcli\.dcli_credstore` directory inside the home directory.	`$HOME/.dcli/.dcli_credstore`
`credstore-add`	Adds login credentials entered for a command to the DCLI credential store file. This option stores the server IP address, session manager, username and password for the command being executed. If an entry already exists, the command updates the entry.	`dcli` directory inside the home directory. `$HOME/dcli`
`credstore-remove`	Removes an entry from the DCLI credential store file. This option removes the entry for a specified server IP address and username if only one session manager is present for a target server and user. In rare cases, information about multiple session manager entries is present. You have to provide the session manager with the `session-manager` option.	
`credstore-list`	Lists all entries stored in the DCLI credential store file. Each entry includes the server IP address, session manager, and user name.	
`session-manager`	Use this option if you use the `credstore-remove` option the same user name and password are stored through multiple session managers. Not usually required.	

Running DCLI Commands

You can display help information for DCLI commands, and run the commands from a system where vCLI is installed, from the vCenter Server Appliance shell, or from a vCenter Server on Windows command prompt.

Displaying Help Information for DCLI Commands

You can display help for each namespace and command by using the `--help` command-line option. Because the available commands depend entirely on the services that are available in the vCenter environment that you are targeting, you must include the server for accurate help information. Help returns the following information for a command:

- Each input option
- Whether the option is required
- Input type

Example

```
dcli com vmware cis tagging tag create --help
usage: com vmware cis tagging tag create [-h] --create-spec-name CREATE_SPEC_NAME
               --create-spec-description CREATE_SPEC_DESCRIPTION --create-spec-category-id
               CREATE_SPEC_CATEGORY_ID

Creates a tag

Input Arguments:
  -h, --help            show this help message and exit
  --create-spec-name CREATE_SPEC_NAME
                        The display name of the tag (required string)
  --create-spec-description CREATE_SPEC_DESCRIPTION
                        The description of the tag (required string)
  --create-spec-category-id CREATE_SPEC_CATEGORY_ID
                        The unique identifier of the parent category in which this tag will be
                        created (required string)
```

Running DCLI Commands Included in the vCLI Package

You run vCLI commands from an administration server on which you installed the vCLI package. After installation, open a command prompt in the `VMware\VMware DCLI` folder, which is at the same level as the VMware vSphere CLI folder.

You specify a server, user name, and password. If you specify `credstore-add`, DCLI creates a credential store file on the local machine and you are no longer required to specify the user name and password when you run DCLI commands again.

Running DCLI Commands on the vCenter Server Appliance

The root user on the vCenter Server Appliance can run DCLI commands from the appliance shell.

1 SSH into the shell or log in to the shell directly as the root user.

   The administrator@vsphere.local user does not have privileges to run DCLI commands.

2 You can run commands individually, or start the interactive DCLI shell. The interactive shell has several advantages including tab completion and a history file.

   ```
   >dcli +interactive
   ```

3 You can list commands, display help for commands, and execute commands. In the example below, the interactive shell uses the `dcli>` prompt, the default.

   ```
   dcli> com vmware vapi metadata cli command list
   ```

Using DCLI with a Credential Store File

To avoid typing in the user name and password each time you run a DCLI command, you can add the current user and the associated password and server IP address to a credential store file by using the `credstore-add` option on the command line.

Passwords are encrypted in the credential store file, but if you want to remove credential store information, you can use `+credstore-remove` to do so.

By default, the credential store file is located in `<homedir>/.dcli/.dcli_credstore`, but you can change the location with the `+credstore-file` option.

Examples

The following examples illustrate how you can interact with the credential store.

1 Add a new credential store entry.

   ```
   dcli com vmware cis tagging tag list +credstore-add +username user1
   ```

2 Remove a credential store entry.

```
dcli +credstore-remove +server <server> +username user1
```

3 List all credential store entries.

```
dcli +credstore-list
```

Order of Precedence for DCLI Authentication

When you run a DCLI command, authentication happens in the order of precedence shown in Table 5-2, "DCLI Authentication Precedence," on page 43. This order of precedence always applies. That means, for example, that you can override an environment variable setting from the command line.

If you are authenticating through vCenter Single Sign-On, the order of precedence is preserved.

Table 5-2. DCLI Authentication Precedence

Authentication	Description
Command line	User name and password, specified on the command line takes precedence, even if a credential store exists.
Environment variable	A user name specified in an environment variable takes precedence over user names in the credential store, but not over the command line.
Credential store	User name and password retrieved from the credential store. A custom credential store file at a non-default location has precedence over a file at the default location.

Input, Output, and Return Codes

DCLI supports the following input arguments.

- **Basic types**. You can enter basic types like string, int, double, or boolean on the command line.
- **List types**. Provide the same option multiple times on the command line and DCLI treats it as a list.
- **Structure**. For structure input fields use the dot notation. For example, id.path is the path field in the id structure.

Currently supported output formatter types are simple, xml, html, table, csv and json. You can change the output format by passing the `formatter` option to DCLI.

For scripting purposes DCLI returns a non-zero error code for an unsuccessful command. To see the last command status in interactive mode, run the `$?` command.

Using DCLI with Variables

You can predefine a set of variables in the environment where you run DCLI commands so you don't have to pass the options each time you run a command. The following environment variables are supported.

Table 5-3. Variables Supported by DCLI

Variable	Description
`DCLI_SERVER`	Set this variable to pass the server IP address. Passing the `server` option on the command line overrides this variable.
`DCLI_SSLCERTFILE`	Set this variable to pass the path of your SSL certificate file. Passing the `ssl-cert-file` option on the command line overrides this variable.
`DCLI_SSLKEYFILE`	Set this variable to pass the path of your SSL key file. Passing the `ssl-key-file` option on the command line overrides this variable.
`DCLI_CACERTFILE`	Set this variable to pass the path of a CA certificate file. Passing the `cacert-file` option on the command line overrides this variable.
`DCLI_USERNAME`	Set this variable to pass the user name required for authentication Passing the `username` option on command line overrides this variable.

Table 5-3. Variables Supported by DCLI

Variable	Description
DCLI_CREDFILE	Set this variable to point to a DCLI credential store file. Default value is ~/.dcli/.dcli_credstore Passing the credstore-file option on the command line overrides variable.
DCLI_HISTFILE	Set this variable to point to a DCLI interactive shell history file path. Default value is ~/.dcli/.dcli_history .
DCLI_HISTSIZE	Set this variable to specify the maximum number of commands to be stored in the DCLI interactive shell history. Default is 500.
DCLI_LOGFILE	Set this variable to specify the log file for DCLI.

DCLI History File

DCLI maintains a history file for each DCLI client that runs in interactive mode. The file stores information on a per-user basis and not on a per-client basis.

Specify the maximum number of commands to be stored with the DCLI_HISTSIZE variable. Default is 500 commands.

You can find the file at the following location:

- Windows: C:\Users\<username>\AppData\VMware\vapi\dcli.log
- vCenter Server Appliance: /var/log/vmware/vapi/dcli.log

Index