Getting Started with vSphere Command-Line Interfaces

17 APR 2018
ESXi 6.7
vCenter Server 6.7

vmware®

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

https://docs.vmware.com/

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 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.

Related Documentation

The documentation for vCLI is available on the VMware {code} vSphere CLI page. Go to https://code.vmware.com/web/dp/tool/vsphere-cli.

- *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 on the VMware {code} PowerCLI 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.

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.

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 (Data Center CLI) 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
- Using ESXCLI for Host Management
- Using PowerCLI to Manage Hosts and Virtual Machines
- Using DCLI to Manage vCenter Services
- vCLI Package Contents

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. You can also manage vCenter services by using DCLI.

Figure 1-1. vSphere CLIs for Host and vCenter Services Management
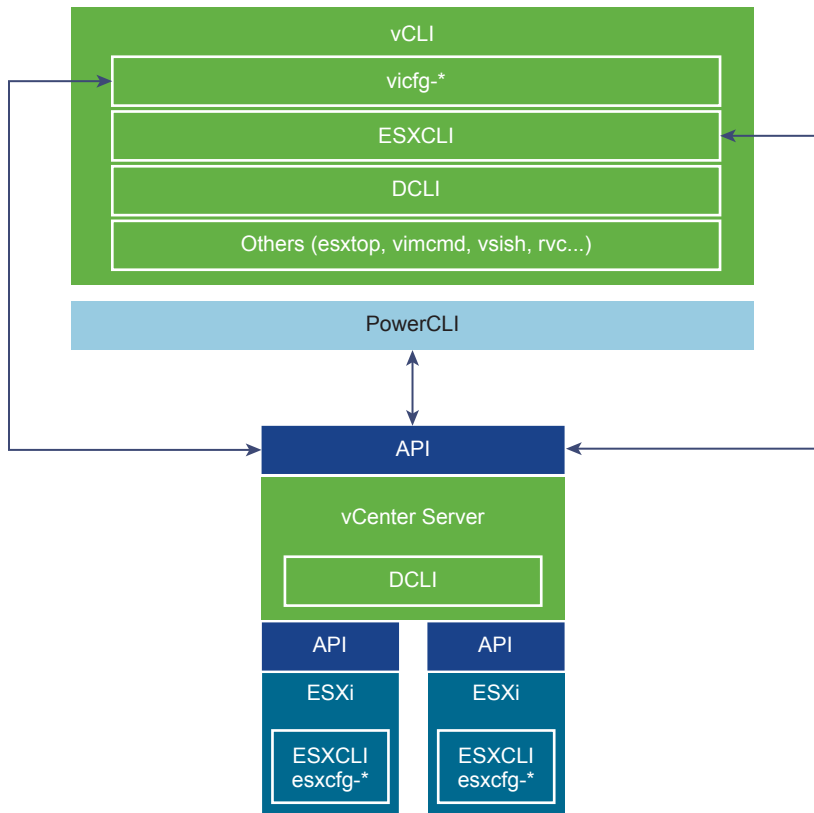
The following command sets are available. For more information about each command set, see the referenced documentation.

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. 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 VMware PowerCLI prompt by using the `Get-EsxCli` cmdlet.	Using ESXCLI for Host Management Chapter 2 Installing vCLI *vSphere Command-Line Concepts and Examples* *vSphere Command-Line Interface Reference*
`vicfg-` and other vCLI commands	Users can manage hosts remotely. Install the vCLI package on a Windows or Linux system, and target the ESXi system that you want to manipulate. `vicfg-` commands are included in this release but are deprecated. Migrate to ESXCLI where possible. 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.	Chapter 2 Installing vCLI *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 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 vSphere Automation SDK interface for managing VMware SDDC services. A DCLI command talks to a vSphere Automation API endpoint to locate relevant information, and then runs the command and displays the 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. ■ 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 Chapter 5 Running DCLI Commands. ■ 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.	Chapter 5 Running DCLI Commands See the *vSphere Automation SDK* documentation for information about supported services and how they interact.
VMware PowerCLI cmdlets	VMware PowerCLI provides a Windows PowerShell interface to the vSphere API. PowerCLI includes PowerShell cmdlets for administering vSphere components. PowerCLI includes more than 600 cmdlets, a set of sample scripts, and a function library for management and automation. The vSphere Image Builder PowerCLI and vSphere Auto Deploy PowerCLI modules are included when you install PowerCLI.	*VMware PowerCLI* documentation
localcli commands	Set of commands for use with VMware Technical Support. localcli commands are equivalent to ESXCLI commands, but bypass the host 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. **Important** If you use a localcli command, an inconsistent system state and potential failure might result.	
pktcap–uw utility	Enables you to monitor the traffic that flows through the physical network adapters, the VMkernel adapters, and the 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 by using commands from 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 *PowerCLI User's Guide* and the *PowerCLI Cmdlet 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 that you are targeting.

ESXCLI Syntax

Each ESXCLI command uses the same syntax.

The following is the standard syntax structure of an ESXCLI command.

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

Syntax Element	Description
`dispatcher options`	Predefined options for connection information such as target host, user name, and so on. See Chapter 4 Running vCLI Host Management Commands. 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 that 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 that you are targeting. See Trust Relationship Requirement for ESXCLI Commands for an example.
`namespace`	Groups ESXCLI commands. vSphere 5.0 and later support nested namespaces.

Syntax Element	Description

command	Reports on or modifies the state of the system.
	
	The following examples show how you can use this element.
	
	`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 of the options displayed in the help or the vCLI reference. For some commands, multiple option values, separated by spaces, are possible. The following example shows how you can use this element. `esxcli system module parameters set -m <module> -p "a=1 b=1 c=1"`

Running ESXCLI Commands Installed as Part of vCLI

You can run an ESXCLI command, installed as part of vCLI, in the ESXi Shell for troubleshooting purposes and remotely against a specific host or against a vCenter Server system.

You can 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 Chapter 2 Installing vCLI.

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 Trust Relationship Requirement for ESXCLI Commands.

See Chapter 4 Running vCLI Host Management Commands.

ESXCLI Command Support When Host and vCLI Versions Do Not Match

When you run an ESXCLI vCLI command, you must know the commands that are supported on the target host specified with `--server`.

The following examples demonstrate command support when versions do not match.

- 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.

- If you run commands against ESXi 6.x hosts, ESXCLI 6.x 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 explore the namespaces for additional help.

Using PowerCLI to Manage Hosts and Virtual Machines

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

PowerCLI is based on Microsoft PowerShell and uses the PowerShell basic syntax and concepts. Microsoft PowerShell is both a command-line and scripting environment. 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 PowerCLI console, by using the following options.

- Through the cmdlet that 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.

See the *PowerCLI User's Guide* on VMware {code}.

Using DCLI to Manage vCenter Services

With the DCLI command set, you can run virtual machine management, appliance management, content library, and tagging commands.

You cannot manage services that are part of vSphere 5.5 or earlier from DCLI. DCLI is not a host management CLI.

DCLI is a CLI client of the vSphere Automation SDK. The following workflow explains how DCLI works.

1   You run a DCLI command.

2   If you are not authenticated, DCLI prompts for a user name and password.

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

4   If you can authenticate, DCLI communicates with the vCenter Server and runs the vSphere Automation API that corresponds to the DCLI command. Different vCenter Server systems support different services.

   Note   If the authenticated user account does not have permissions to run the DCLI command, you receive an `Unauthorized` error message, even if the user credentials are correct.

5   DCLI displays the result or an error message.

You can run DCLI commands as follows.

- vCLI package - Install the vCLI package on the server of your choice. You can then run DCLI commands against an endpoint. See Using DCLI Commands.

- vCenter Server Appliance - Run DCLI commands from the vCenter Server Appliance shell. See Running DCLI Commands on the vCenter Server Appliance.

- 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 DCLI connection and formatting options, each preceded by a plus (+) sign. You also specify the namespace, the command, and the command options. Namespaces 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 two minus signs (−−).

The syntax of a DCLI command is the following.

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

The following table describes the DCLI syntax elements.

Syntax Element	Description
DCLI options	Predefined options for connection information including the vSphere Automation SDK endpoint and formatting options. Always preceded by a plus (+) 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 vSphere Automation SDK namespaces and are nested.
command	Reports on or modifies the state of the system.
option and value	Command option and value pairs preceded by two minus signs (−−).

Examples

```
./dcli +server myvc com vmware vcenter vm list
```

```
./dcli com vmware vmc orgs sddcs list −−org orgID +vmc−server
```

Note   Under Windows, you can navigate to the directory that contains the DCLI binary and start commands with `dcli` instead of `./dcli`.

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 or VMware Cloud on AWS. 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 - These commands are available as part of vCLI, from the vCenter Server 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. `vicfg-` commands are still included in this release but are deprecated. Migrate to ESXCLI where possible.

    - 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 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 Chapter 2 Installing vCLI.

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 or host name 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 Chapter 4 Running vCLI Host Management Commands and Using DCLI Commands.

Installing vCLI

2

You can install a vCLI package on a Linux or a Microsoft Windows system.

This chapter includes the following topics:

- Installation Overview
- Overview of Linux Installation Process
- Installing and Uninstalling vCLI on Linux Systems
- Installing and Uninstalling vCLI on Windows
- Enabling Certificate Verification

Installation Overview

You can install a vCLI package on a supported platform.

You can install a vCLI package on a physical or virtual machine. See Installing and Uninstalling vCLI on Linux Systems and Installing and Uninstalling vCLI on Windows.

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 content of the installer package differs for different platforms.

Platform	Installation Process
Windows	You must install required software. The installation package includes vCLI and vSphere SDK for Perl.
Linux	You must install required software and you must have Internet access. See Installing Prerequisite Software for Linux Systems. 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 can specify the target server connection options directly or indirectly. You can also write scripts and manage your vSphere environment using those scripts.

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 software 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 Prerequisite Software for Linux Systems.
LibXML2	Used for XML parsing. The vCLI client requires 2.9.6 or later. If you have an older version installed, you must upgrade to 2.9.6 or later.
	The `libxml2` package is not included in the default Linux distribution. See Installing Prerequisite Software for Linux Systems.
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 Prerequisite Software for Linux Systems for instructions.

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

- `Crypt-SSLeay-0.72`

- `IO-Compress-Zlib-2.037`

- `Compress-Raw-Zlib-2.037`

- `Archive-Zip-1.28`

- `Data-Dumper-2.121`

- `XML-LibXML-2.0129`

- `libwww-perl-6.15`

- `LWP-Protocol-https-6.07`

- `XML-LibXML-Common2.0129`

- `XML-NamespaceSupport-1.12`

- `XML-SAX-0.99`

- `UUID-0.27`

- `SOAP-Lite-0.710.08`

- `HTML-Parser-3.60`

- `version-0.78`

- `MIME-Base64-3.14`

- `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 must install that module.

Note If you intend to run vCLI commands with SSL certification, verify 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 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 and Uninstalling vCLI on Linux Systems.

- If an earlier version of a recommended module is found, the installer does not install a different version from CPAN and proceeds with the installation. After completing the 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 by using the package installer for your platform, the installation CD, or CPAN.

- If a later version of a recommended module is found, the installer proceeds with the installation and does not display a message after the installation.

Note The installer does not overwrite existing versions of recommended Perl modules. You must update those modules manually.

5   After all required software and all prerequisite Perl modules are installed, you can install vCLI. See Installing Prerequisite Software for Linux Systems.

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 reset the variable, before or after the installation, by using the command appropriate for your distribution and shell, for example `setenv` or `export`. If you do not reset the path, the system might still look for executable files in the old location.

Installing and Uninstalling vCLI on Linux Systems

Before you can install the vCLI package on a Linux system, that system must meet specific 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 Prerequisite Software for Linux Systems

If the 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.

Platform	Installation

RHEL 6.6 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```

RHEL 7.1 64-bit	Find the required modules on the installation DVD, or use yum to install them.
	
	```yum install e2fsprogs-devel libuuid-devel openssl-devel perl-devel```
	```yum install glibc.i686 zlib.i686```
	```yum install perl-XML-LibXML libncurses.so.5 perl-Crypt-SSLeay```

RHEL 7.3 64-bit	Find the required modules on the installation DVD, or use yum to install them.
	
	```yum install e2fsprogs-devel libuuid-devel openssl-devel perl-devel```
	```yum install glibc.i686 zlib.i686```
	```yum install perl-XML-LibXML libncurses.so.5 perl-Crypt-SSLeay```
	
	You might have to install GCC by running `yum group install "Development Tools"` if there is no compiler already installed.

SUSE Enterprise	Install the prerequisite packages from the SLES SDK DVD. When you insert the DVD, it offers to autorun. Cancel the autorun and use the `yast` package installer to install OpenSSL or other missing required packages.
	
	- SLES 11 SP3 64-bit
	
	`yast -i openssl-devel libuuid-devel libuuid-devel-32bit`
	
	- SLES 12 64-bit
	
	`yast -i openssl-devel libuuid-devel libuuid-devel-32bit e2fsprogs-devel`
	
	Some users might be authorized to use the Novell Customer Center and use `yast` to retrieve missing packages from there.

Platform	Installation
Ubuntu 12.04 64-bit	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. ``` 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 12.04 64-bit, the `resxtop` and ESXCLI commands do not work if you do not install the 32-bit compatibility libraries.
Ubuntu 14.04 64-bit	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. ``` sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0 gcc-multilib build- essential gcc uuid uuid-dev perl libssl-dev perl-doc liburi-perl libxml-libxml-perl libcrypt-ssleay-perl ``` For Ubuntu 14.04 64-bit, the `resxtop` and ESXCLI commands do not work if you do not install the 32-bit compatibility libraries.
Ubuntu 15.10 64-bit	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. ``` sudo apt-get install lib32z1 lib32ncurses5 build-essential uuid uuid-dev perl libssl-dev perl-doc libxml-libxml-perl libcrypt-ssleay-perl libsoap-lite-perl ``` For Ubuntu 15.10 64-bit, the `resxtop` and ESXCLI commands do not work if you do not install the 32-bit compatibility libraries.
Ubuntu 16.04 64-bit	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. ``` sudo apt-get install lib32z1 lib32ncurses5 build-essential uuid uuid-dev libssl-dev perl-doc libxml-libxml-perl libcrypt-ssleay-perl libsoap-lite-perl libmodule-build- perl ``` For Ubuntu 16.04 64-bit, the `resxtop` and ESXCLI commands do not work if you do not install the 32-bit compatibility libraries.

Install the vCLI Package on a Linux System

You can install the vCLI package and run a command to verify that installation was successful.

Prerequisites

Verify that you have installed the required prerequisite software.

Procedure

1 Download the vCLI Linux 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 Log in as root.

3 Untar the vCLI binary that you downloaded.

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

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

4 (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
   ```

5 Run the installer.

   ```
   sudo vmware-vsphere-cli-distrib/vmware-install.pl
   ```

6 To accept the license terms, enter **yes** and press Enter.

   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`

What to do next

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.

Uninstall the vCLI Package on Linux

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

Procedure

1   Navigate to the directory where you installed vCLI.

    The default directory 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.

Install the vCLI Package on Windows

The vCLI installation package for Windows installs vSphere SDK for Perl and vSphere CLI, but does not include the ActivePerl runtime from ActiveState Software.

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.

Prerequisites

Verify that you have ActivePerl or Strawberry Perl version 5.14 or later installed on your Windows system.

Procedure

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 an alternative 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.

If you do not reboot, path settings might not be correct on your Windows platform.

Uninstall the vCLI Package on Windows

You can uninstall the vCLI package by following the standard Windows procedure.

Procedure

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

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

3   Click **Yes** when prompted.

The system uninstalls vCLI and vSphere SDK for Perl.

Enabling Certificate Verification

You can enable certificate verification by using variables.

The vSphere SDK for Perl and vCLI use `Crypt::SSLEay` to support certificate verification. `Crypt::SSLEay` enables 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 display error or warning messages. Use *HTTPS_DEBUG* for troubleshooting before running vCLI commands.

Running Host Management Commands in the ESXi Shell

3

Usually, 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.

To run commands, you must first establish access to the ESXi Shell.

This chapter includes the following topics:

- ESXi Shell Access with the Direct Console
- Remote ESXi Shell Access with SSH
- Lockdown Mode
- Run an ESXCLI Command in the ESXi Shell

ESXi Shell Access with the Direct Console

An ESXi system includes a Direct Console User Interface (DCUI) that you can use to start and stop the system and to perform a limited set of maintenance and troubleshooting tasks.

You can use the direct console to access 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.

- With local shell access, you can log in to the shell directly from the Direct Console. See Enabling Local ESXi Shell Access.
- With remote shell (SSH) access you can connect to the host by 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.

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.

You can install the vCLI package on a supported Windows or Linux system, 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 the vSphere Web Client.

Enable the ESXi Shell in the Direct Console

If you have access to the Direct Console Interface, you can enable the ESXi Shell from there.

Procedure

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   Select **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.

What to do next

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

Enable the ESXi Shell from the vSphere Web Client

If you do not have access to the Direct Console Interface, you can enable the ESXi Shell from the vSphere Web Client.

Procedure

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**.

What to do next

After you enable the ESXi Shell, you can use it 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 or from the vSphere Web Client. See the *vSphere Security* document for detailed instructions.

Use the Local 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.

Procedure

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 enter the password, characters are not displayed on the console.

3 Enter shell commands to perform management tasks.

4 To log out, enter `exit` in the shell.

5 To return to the direct console, press Alt-F2.

What to do next

See the *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.

Enable SSH Access in the Direct Console

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

Prerequisites

Procedure

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.

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 menu.

What to do next

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

Enable SSH from the vSphere Web Client

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

Procedure

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**.

What to do next

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

Access the Remote ESXi Shell with SSH

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

Procedure

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 more fine-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 by using the Add Host wizard to add a host to vCenter Server, by using the vSphere Web Client to manage a host, or by using the Direct Console User Interface (DCUI).

See the *vSphere Security* documentation for details on lockdown mode in vSphere 6.x.

Run an ESXCLI Command in the ESXi Shell

You can run ESXCLI commands in the ESXi Shell 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.

Prerequisites

Verify that the ESXi Shell is enabled.

Procedure

1 Log in to the shell.

2 Run the command.

    For example, to list NFS storage devices, run the following command.

    ```
    esxcli storage nfs list
    ```

What to do next

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 vCLI host management commands from the command line of the system where you installed the package and from scripts.

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

Note `vicfg-` commands are deprecated. A deprecation note is displayed when you run these commands. You can use ESXCLI equivalents of `vicfg-` commands.

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.

Note See Chapter 5 Running DCLI Commands for information about DCLI commands, which you can use to manage vCenter Server services.

This chapter includes the following topics:

- Overview of Running vCLI Host Management Commands
- Protecting Passwords
- Authenticating Through vCenter Server and vCenter Single Sign-On
- Authenticating Directly to the Host
- Trust Relationship Requirement for ESXCLI Commands
- Common Options for vCLI Host Management Command Execution
- Using vCLI Commands in Scripts
- Run Host Management Commands from a Windows System
- Run Host Management Commands from a Linux System

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 or by running 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.

- Prepare scripts that contain vCLI commands. Then run the scripts from a system that has the vCLI package installed. See Using vCLI Commands in Scripts.

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.

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

- Specify the vCenter Server system with the `--server` option and the ESXi host with the `--vihost` option.

- Specify only the ESXi host with the `--vihost` option.

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.

Protecting Passwords

You can follow different password protection approaches depending on your environment setup.

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 that you enter.

- For noninteractive use, you can create a session file using the `save_session` option. See Create and Use a Session File.

- 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.

- Use variables or configuration files.

- If you are running vCLI 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.

With vCLI, you can 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, even the root user cannot run vCLI commands directly against ESXi hosts in lockdown mode. See vCLI and Lockdown Mode and the *vSphere Security* documentation.

Order of Precedence for vCLI Host Management Commands

When you run a vCLI host management command, authentication happens in order of precedence.

The order of precedence is described in the following table. 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 to vCenter Server.

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.

Authentication	Description	See
Command line	Password (`--password`), session file (`--sessionfile`), or configuration file (`--config`) specified on the command line.	Create and Use a Session File
Environment variable	Password specified in an environment variable.	Using Environment Variables
Configuration file	Password specified in a configuration file.	Using a Configuration File
Current account (Active Directory)	Current account information used to establish an SSPI connection. Available only on Windows.	Using the Microsoft Windows Security Support Provider Interface
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.

The 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 the following 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.

Create and Use a Session File

You can create a session file with the `save_session` script.

The `save_session` 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.

Procedure

1   Navigate to the directory where the script is located.

The following examples show possible locations of the script.

Operating System	Command
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 which to connect to and the name of a session file in which the script saves an authentication cookie.

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

The following examples show the correct syntax for each OS.

Operating System	Command
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>`

Note   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>
```

The following examples show the correct syntax for each OS.

Operating System	Command
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

How you use environment variables depends on the operating system that you are using.

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 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 Common Options for vCLI Host Management Command Execution.

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

Pass in the configuration file when you run vCLI commands, by using the following syntax.

```
<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 by 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$.

Example: 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
```

Example: 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 the Microsoft Windows Security Support Provider Interface

With the `--passthroughauth` option, which is available if you run vCLI commands from a Microsoft Windows system, you can use the Microsoft Windows Security Support Provider Interface (SSPI).

You can refer to the Microsoft Web site for detailed information on 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 that 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.

```
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
```

This example establishes a connection 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.

Download and Install the vCenter Server Certificate

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

Procedure

1   Enter the URL of the vCenter Server system or vCenter Server 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.

    A certificates folder is extracted. 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 that you are on.

What to do next

You can now run ESXCLI commands against any host that is managed by the trusted vCenter Server system without supplying additional information if you specify the vCenter Server system 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 ESXi host or vCenter Server system in the `--thumbprint` parameter or the *VI_THUMBPRINT* variable.

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

```
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
```

Use the Credential Store

Your vCLI installation includes a credential store. You can establish trust for a user with the 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.

Procedure

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 must pass it in with the `--credstore` option.

If you do not use the `--credstore` option, the host becomes accessible without authentication.

Common Options for vCLI Host Management Command Execution

You can use connection options that are available for all vCLI host management commands and common options that you can use when you run a `vicfg-` vCLI command.

vCLI Connection Options

The following table 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 Run Host Management Commands from a Windows System and Run Host Management Commands from a Linux System.

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 which encoding to use. Several encodings are supported. ■ `utf8` ■ `cp936` (Simplified Chinese) ■ `shftjis` (Japanese) ■ `iso-885901` (German) You can use `--encoding` to specify the encoding for vCLI to map to when it is run on a foreign language system.

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 detailed information on 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 by 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 is idle 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.
––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.

Option and Environment Variable	Description
`--url <url>` `VI_URL=<url>`	Connects to the specified vSphere Web Services SDK URL.
`--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`.

vCLI Common Options

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

Option	Description
`--help`	Prints a brief usage message. The message first lists 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
```

Run 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.

Procedure

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>
    ```

    The following examples show the difference between ESXCLI and `vicfg-` syntax.

    ```
    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.

Run 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.

Procedure

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. The following examples show the difference between ESXCLI and `vicfg-` syntax.

```
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. 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 enables you to run some commands without additional authentication. However, for management of certain services, you might be prompted for a user name and password.

This chapter includes the following topics:

- Overview of Running DCLI Commands

- Using DCLI Commands

- DCLI Authentication

- Using the DCLI Configuration Store

- DCLI Modules

- Internal DCLI Commands

- Using Default Options

- Input, Output, and Return Codes

- Using DCLI with Variables

- DCLI Security

- Setting the DCLI Log File

- DCLI History File

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 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.

- 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 supports running commands in either interactive or non-interactive mode.

- Interactive mode allows for running commands quickly.

- Non-interactive mode is suitable for scripting purposes.

DCLI Syntax

Each DCLI command uses the same syntax.

The command name can be followed by DCLI connection and formatting options, each preceded by a plus (+) sign. You also specify the namespace, the command, and the command options. Namespaces 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 two minus signs (−−).

The syntax of a DCLI command is the following.

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

The following table describes the DCLI syntax elements.

Syntax Element	Description
DCLI options	Predefined options for connection information including the vSphere Automation SDK endpoint and formatting options. Always preceded by a plus (+) 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 vSphere Automation SDK namespaces and are nested.
command	Reports on or modifies the state of the system.
option and value	Command option and value pairs preceded by two minus signs (−−).

Examples

```
./dcli +server myvc com vmware vcenter vm list
```

```
./dcli com vmware vmc orgs sddcs list --org orgID +vmc-server
```

Note Under Windows, you can navigate to the directory that contains the DCLI binary and start commands with `dcli` instead of `./dcli`.

DCLI Options

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

For many of the options, you can instead use variables. See Using DCLI with Variables.

```
dcli [+server SERVER_IP]
     [+vmc-server [VMC_SERVER]]
     [+interactive]
     [+prompt PROMPT]
     [+skip-server-verification]
     [+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 ...]]
     [+configuration-file CONFIGURATION_FILE]
```

With these options, you can provide the following information. If you are entering options in DCLI interactive mode, drop-down autocompletion is supported on Linux and Window systems. In all cases, you can specify a partial option if the option is not ambiguous. For example, +i indicates interactive, but you have to specify, at least, +credstore-a to disambiguate `option +credstore-add`.

The following table describes the DCLI options.

Option	Description	Default	Available in Interactive Mode
server	The vCenter Server system to which DCLI connects.	localhost	No
vmc-server	The VMware Cloud on AWS server to which DCLI connects.	https://vmc.vmware.com	No

Option	Description	Default	Available in Interactive Mode
interactive	Runs DCLI in interactive shell mode, which supports drop-down autocompletion 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.		No
prompt	Prompt that the interactive shell uses.	dcli>	No
skip-server-verification	Skips the server SSL verification process.	False	No
cacert-file	Specifies the certificate authority certificates for validating SSL connections.		No
more	Displays page-wise output.		Yes
formatter	Output formatter, which has one of the following possible values. ■ simple ■ table ■ xml ■ json ■ html ■ csv	Default is table for lists of structures and simple for all other output including json.	Yes
loglevel	The log level, which has one of the following possible values. ■ debug ■ info ■ warning ■ error	info	Yes
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. **Note** Only available when DCLI connects to a vCenter Server system.		Yes

Option	Description	Default	Available in Interactive Mode
password	You can use this option to provide your password explicitly instead of waiting for the DCLI password prompt. **Note** Only available when DCLI connects to a vCenter Server system.		Yes
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 filename does not work in your environment. By default, the credential store file is in the `.dcli/.dcli_credstore` directory inside the home directory.	`$HOME/.dcli/.dcli_credstore`	Yes
credstore-add	Adds login credentials entered for a command to the DCLI credential store file. This option stores the server IP address, session manager, user name, and password for the command being run. If an entry exists, the command updates the entry.		Yes
credstore-remove	Removes an entry from the DCLI credential store file. This option removes the entry for a specified server IP address and user name if only one session manager is present for a target server and user. In rare cases, information about multiple session manager entries is present. You must provide the session manager with the `session-manager` option.		Yes
credstore-list	Lists all entries stored in the DCLI credential store file. Each entry includes the server IP address, session manager, and user name.		Yes
session-manager	Use this option if you use the `credstore-remove` option the same user name and password are stored through multiple session managers. Not usually required.		Yes
configuration-file	Path to the configuration store file to use for options and profile-related internal commands.	`$HOME/.dcli/.dcli_configuration`	No

Run DCLI Commands in Interactive Mode

DCLI supports interactive shell mode which you can activate by using `dcli +interactive`.

Interactive mode supports drop-down autocompletion of namespaces, commands, command options, and option values in case of enumeration values. With DCLI interactive mode, you can also pass a short command if it is uniquely resolvable. For example, `dcli> com vmware vmc vm list` can also be run as `dcli> vm list`.

Interactive mode is also a quicker way to browse various namespaces and commands, as DCLI caches the list of namespaces and commands available on the server for faster access. DCLI interactive mode provides specific shell commands which can be accessed by running `dcli> help`.

You can change the prompt for DCLI interactive mode by using `dcli +interactive +prompt <user-prompt>` when entering interactive mode.

Procedure

1 From the command line, navigate to the location of the DCLI binary.

2 Enable interactive mode.

```
./dcli +vmc +i
```

DCLI connects to the VMware Cloud on AWS server in interactive mode.

3 List all SDDCs in a specified organization.

- Provide the full interactive mode command.

```
dcli> com vmware vmc orgs sddcs list --org orgID
```

- Provide the short interactive mode command.

```
dcli> orgs sddcs list --org orgID
```

Note Both commands should return the same result.

Run DCLI Commands in Non-Interactive Mode

When you run DCLI in non-interactive mode, you must provide the full command, including the DCLI options.

Procedure

1 From the command line, navigate to the location of the DCLI binary.

2 List all SDDCs in a specified organization.

```
./dcli +vmc com vmware vmc orgs sddcs list --org orgID
```

Using 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. For VMware Cloud on AWS, the available commands depend on the services that are available in the REST API that you are targeting.

Help returns the following information for a command.

- Each input option

- Whether the option is required

- Input type

Examples

```
dcli com vmware cis tagging tag create --help
usage: com vmware cis tagging tag create [-h] --name NAME --description DESCRIPTION --category-id
CATEGORY_ID

Creates a tag

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

```
dcli> com vmware vmc orgs sddcs list --help
usage: com vmware vmc orgs sddcs list [-h] [--org ORG]

Lists all SDDCs of an organization

Input Arguments:
  -h, --help  show this help message and exit
  --org ORG   Organization identifier (required) (string)
```

Running DCLI Commands Included in the vCLI Package

You can run vCLI commands from an administration server on which you installed the vCLI package.

After installation, the VMware DCLI folder is available inside your vCLI installation path. You must open a command prompt in your installation folder path and navigate to the VMware DCLI folder.

You must specify a server, user name, and password. If you specify `credstore-add`, DCLI creates a credential store file on the local machine. As a result, 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.

The following options are available.

- Use SSH to connect to the shell or log in to the shell directly as the root user.

- You can run commands individually, or start the interactive DCLI shell. The interactive shell has several advantages including drop-down autocompletion and a history file.

```
>dcli +interactive
```

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

```
dcli> com vmware vcenter vm list
```

DCLI Authentication

Most DCLI commands require authentication. VMware Cloud on AWS and vCenter Server use different authentication mechanisms.

VMware Cloud on AWS uses a refresh token for authentication. vCenter Server requires credentials, which you can provide in different ways.

VMware Cloud on AWS Authentication

You can authenticate to VMware Cloud on AWS by using a refresh token.

You can obtain a refresh token from your VMware Cloud on AWS user profile under the Access Tokens area, or from the VMware Cloud on AWS administrators in your organization. After you receive a refresh token, you can provide it to DCLI when prompted. DCLI prompts for a refresh token when you try to access VMware Cloud on AWS.

```
./dcli +vmc +i
Refresh Token:
```

The refresh token is provided in a secure way. Be careful that you are using the copy/paste functionality of your terminal correctly to avoid pasting the refresh token more than once. After you provide the refresh token, DCLI asks whether you want to save it to the credential store. If you choose to save the refresh token, you will not need to provide it each time you connect to VMware Cloud on AWS.

Updating the Refresh Token

The following example illustrates how you can update your VMware Cloud on AWS refresh token.

1 Remove the old refresh token.

```
./dcli +vmc +credstore-remove
```

2 Connect to VMware Cloud on AWS and provide the new refresh token when prompted.

```
./dcli +vmc
Refresh Token:
```

vCenter Server Authentication

You can authenticate to vCenter Server by providing a user name and password.

You can provide credentials in different ways.

■ Specify the +username option.

```
dcli> vcenter vm list +username user42 +password
```

Note If you skip the +password option, you can provide the password in a more secure way when prompted.

■ Specify the *DCLI_USERNAME* environment variable.

```
export DCLI_USERNAME=user42
./dcli +server vcenter_server_IP com vmware vcenter vm list
```

■ Provide credentials when prompted by DCLI.

Note You are prompted for credentials if the operation requires authentication.

```
dcli> vcenter vm list
Username: user42
Password:
Do you want to save credentials in the credstore? (y or n) [y]:
```

■ Save the credentials in the credential store.

Note   You must provide the command against which the authentication should be applied.

```
./dcli +server vcenter_server_IP +skip +credstore-add +username user42 +password com vmware
vcenter vm list
```

After you save the user name and password in the credential store, you will not need to provide credentials each time you connect to vCenter Server.

Using DCLI with a Credential Store File

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

Passwords and refresh tokens are encrypted in the credential store file. 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 `$HOME/.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.

■ Add a new credential store entry.

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

■ Remove a credential store entry.

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

■ List all credential store entries.

```
./dcli +credstore-list
```

Order of Precedence for DCLI Authentication to vCenter Server

When you run a DCLI command, authentication happens in order of precedence, which 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.

The following table shows the DCLI authentication precedence order.

Authentication	Description
Command line	The user name and password specified on the command line take 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	The 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.

Using the DCLI Configuration Store

The configuration store is a store for key-value tuples. It stores simple key-value items in dictionary-like way.

The configuration store has profiles. Profiles isolate key-value pairs. DCLI uses a default profile. You can specify the default profile by using the `default_profile` key.

In the following example, the default profile is labeled `default`. Each profile first specifies key-value items for a given module. The example contains the `vmc` and `vsphere` modules. For more information about modules, see DCLI Modules. The profile then specifies the server you want the key-value pair to be valid for, and also that this pair should be used for the default options DCLI functionality. For more information about default options, see Using Default Options.

Example: Configuration Store Syntax

```
{
    "configuration": {
        "version": "1.0",
        "profiles": {
            "default": {
                "vmc": {
                    "https://vmc.vmware.com": {
                        "default_options": {
                            "org": "myOrgId"
                        }
                    }
                },
                "vsphere": {
                    "https://myVsphereIp": {
                        "default_options": {
                            "vm": "myVMID"
                        }
                    }
                },
            }
        },
        "default_profile": "default"
    }
}
```

Configuration Store Path

The default DCLI configuration store path is `$HOME/.dcli/.dcli_configuration`. You can set a different configuration store path by using either the *DCLI_CONFIGFILE* environment variable or the `+credstore-file` option.

DCLI Modules

DCLI modules represent encapsulation of API commands.

There are two modules in DCLI labeled `vmc` and `vsphere`. The `vmc` module becomes active when you connect to VMware Cloud on AWS, and the `vsphere` module becomes active when you connect to vCenter Server.

Internal DCLI Commands

DCLI provides specific internal commands. You can use internal commands to alter the configuration store, active profile, and default options.

Internal commands are accessible through the first level of the `env` namespace. The following table lists all internal commands.

Internal Command	Description
`env profiles default set`	Sets the default profile.
`env profiles default get`	Retrieves the current default profile.
`env profiles add`	Adds a new profile to the configuration store.
`env profiles get`	Retrieves information about the specified profile.
`env profiles list`	Lists all available profiles.
`env profiles delete`	Deletes the specified profile.
`env options set`	Sets the specified default option to a specified value.
`env options get`	Retrieves the specified default option.
`env options delete`	Deletes the specified default option.
`env options list`	Lists all default options.

Using Default Options

You can use default options to provide predefined values for specific options.

The following examples show the full commands, which contain a complex value for the `org` parameter.

```
dcli> com vmware vmc sddcs list --org someReallyLongOrgID
```

```
dcli> com vmware vmc orgs get --org someReallyLongOrgID
```

By using default options, you can set the `org` parameter as a default option and skip adding it to the command.

```
dcli> env options set --option org --value someReallyLongOrgID
```

Performing this operation allows you to run the same commands without specifying the parameter value and still receive the same results.

```
dcli> com vmware vmc sddcs list
```

```
dcli> com vmware vmc orgs get
```

If you want to provide a value for the `org` parameter that is different to the value set for the default option, you can specify it manually.

```
dcli> com vmware vmc sddcs list --org anotherReallyLongOrgID
```

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	You can provide the same option multiple times on the command line and DCLI treats it as a list.

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 do not have to pass the options every time you run a command. The following environment variables are supported.

Variables Supported by DCLI

Variable	Description
DCLI_VMC_SERVER	Set this variable to pass the VMware Cloud on AWS server IP address.
DCLI_SERVER	Set this variable to pass the vCenter Server system IP address.
	Passing the `server` option on the command line overrides this variable.

Variable	Description
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.
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_LOGFILE	Set this variable to specify the log file for DCLI.
DCLI_CACERTS_BUNDLE	Set this variable to specify the path to a trust store. By default, DCLI includes a trust store from Certifi.
DCLI_VMC_METADATA_URL	Set this variable to specify metadata URL location. By default, DCLI receives metadata from VMware Cloud on AWS for API specifics such as available commands.
DCLI_VMC_METADATA_FILE	Set this variable to specify metadata file location. If set, DCLI retrieves metadata from the specified file.
DCLI_COLOR_THEME	Set this variable to change the color output of DCLI. The possible values are **empty**, **blue**, **red**, **green**, **yellow**, **gray**.
DCLI_CONFIGFILE	Set this variable to specify a custom configuration file.

DCLI Security

DCLI enhances security by providing SSL communication and secret input functionality.

SSL Communication

By default, DCLI verifies SSL certificates for HTTPS requests and throws an error message if it is unable to verify the certificate.

DCLI bundles certificates from `certifi.io` by using the `cacert.pem` file provided in the DCLI package. If you want to override this certification method, you have the following options.

- Use the *DCLI_CACERTS_BUNDLE* environment variable to specify a path to a custom certificates store file.

- Use the `+cacert-file` DCLI option to specify a path to a custom certificates store file.

- Use the `+skip-server-verification` DCLI option to skip certificates verification.

  Note Skipping certificates verification presents a security risk.

DCLI Secret Input

Some command options, more commonly when connecting to vCenter Server, are of secret type. Values for command options of secret type should be provided in a secure way.

You can provide values for secure options in two ways.

- Provide the value explicitly in the command line.

```
dcli> com vmware vcenter securecommand -secureoption securevalue
```

- Provide the value when prompted after initially skipping the value.

```
dcli> com vmware vcenter securecommand -secureoption
secureoption:
```

Setting the DCLI Log File

You can set the DCLI log file, which can trace verbose log information, by using the *DCLI_LOGFILE* environment variable.

If you want to see additional log information in the console, you can use the `+loglevel debug` option as shown in the following example.

```
./dcli +vmc +loglevel debug
```

Note   DCLI does not preserve any sensitive information, such as passwords and secure input, in the log file.

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.

You can find the file at the following location.

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