Getting Started with ESXCLI

VMware vSphere 8.0
VMware ESXi 8.0
vCenter Server 8.0

vmware
by Broadcom

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

https://docs.vmware.com/

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

Contents

About This Book

Getting Started with ESXCLI gets you started with ESXi Shell commands and ESXCLI commands.

Intended Audience

This book is for experienced system administrators who are familiar with vSphere administration tasks.

Overview of ESXCLI

1

You can use ESXCLI to manage many aspects of an ESXi host by running ESXCLI commands remotely or in the ESXi Shell.

You can run ESXCLI commands against a vCenter Server system and target the host indirectly. Running against vCenter Server systems by using the `--vihost` option is required if the host is in lockdown mode. You can also run ESXCLI commands in the local ESXi Shell to manage that host.

Read the following topics next:

- Using ESXCLI for Host Management
- ESXCLI Syntax
- Running ESXCLI Commands
- ESXCLI Command Support When Host and ESXCLI Versions Do Not Match

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 remotely, 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 *ESXCLI 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 [connection options] <namespace> [<namespace> ...] <cmd> [cmd options]
```

Syntax Element	Description
`connection options`	Predefined options for connection information such as target host, user name, and so on. See Chapter 4 Running ESXCLI 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** 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 the following 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. Nested namespaces are supported.

`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 ESXCLI 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

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

You can install ESXCLI on one of the supported Windows, Linux, or macOS systems. 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 ESXCLI.

Note 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 ESXCLI Host Management Commands.

ESXCLI Command Support When Host and ESXCLI Versions Do Not Match

When you run an ESXCLI 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 6.x hosts, ESXCLI 6.x commands are supported.
- If you run commands against ESXi 7.x hosts, ESXCLI 7.x commands are supported.
- If you run commands against ESXi 8.x hosts, ESXCLI 8.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 host. You can explore the namespaces for additional help.

Run `esxcli --server <target> esxcli command list` for a list of all commands supported on the target host.

Installing ESXCLI

2

You can install ESXCLI on a Linux, Microsoft Windows, or a macOS system.

Read the following topics next:

- Installing and Uninstalling ESXCLI

Installing and Uninstalling ESXCLI

You can install and uninstall ESXCLI by running Python commands.

You must have a compatible version of Python installed on a supported operating system. The supported Python versions are listed in the *Release Notes*.

Install ESXCLI

You can install ESXCLI by downloading the installation package and running a Python command.

Prerequisites

- Verify that there is no existing ESXCLI installation on the system.

Procedure

1. Download the ESXCLI Python package.

   You can find a link to the package on the ESXCLI page, which you can find in the **Developer Tools** section of the **Resources** tab on the VMware Developer website.

2. Navigate to the folder where you downloaded the package and run the Python `pip` installer.

```
$ pip install esxcli-8.0.0-XXXXXXXX.tgz
```

Uninstall ESXCLI

You can uninstall ESXCLI by running a Python command.

Procedure

- Run the uninstaller.

```
$ pip uninstall esxcli
```

Running Host Management Commands in the ESXi Shell 3

Usually, installing ESXCLI 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.

Read the following topics next:

- 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 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 ESXCLI package.

You can install the ESXCLI 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 Client.

Enable the ESXi Shell in the Direct Console

If you have access to the Direct Console User 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 Client

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

Procedure

1. In the vSphere Client, select the host
2. On the **Configure** tab, select **System > Services**.
3. Select **ESXi Shell**.
   - To temporarily start or stop the service, click the **Start** or **Stop** button.
   - To change the startup policy, click the **Edit Startup Policy** button.

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 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, type **`exit`** in the shell.
5. To return to the direct console, press Alt-F2.

What to do next

See the *ESXi 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

If running commands remotely is disabled on an ESXi host, you cannot log in to the host by using a remote shell. You can enable running commands remotely from the direct console or from the vSphere 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 Client

If running commands remotely is disabled on an ESXi host, you cannot log in to the host by using a remote shell. You can enable running command remotely from the direct console or from the vSphere Client.

Procedure

1. In the vSphere Client, select the host
2. On the **Configure** tab, select **System > Services**.
3. Select **SSH**.
   - To temporarily start or stop the service, click the **Start** or **Stop** button.
   - To change the startup policy, click the **Edit Startup Policy** button.

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.

- 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. You can also use the vSphere 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 ESXCLI 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 Client to manage a host, or by using the Direct Console User Interface (DCUI).

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

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 ESXCLI Host Management Commands

4

You can run ESXCLI host management commands from the command line of the system where you installed the package and from scripts.

Host management 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.

Important If an ESXi system that you target is in lockdown mode, you cannot run ESXCLI 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 ESXCLI and Lockdown Mode.

Read the following topics next:

- Overview of Running ESXCLI 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
- Connection Options for ESXCLI Host Management Command Execution
- Using ESXCLI Commands in Scripts
- Run Host Management Commands from a Windows System
- Run Host Management Commands from a Linux System

Overview of Running ESXCLI Host Management Commands

You can run ESXCLI 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 ESXCLI or by running scripts.

- Open a command prompt on a Linux or Windows system on which you installed ESXCLI. Enter commands into that command prompt, specifying connection options. See Authenticating Directly to the Host.
- Prepare scripts that contain ESXCLI commands. Then run the scripts from a system that has the ESXCLI package installed. See Using ESXCLI Commands in Scripts.

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 ESXCLI 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 non-interactive use, you can create a session file by using the `save_session` option. See the *vSphere SDK for Perl Programming Guide*.

- 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 ESXCLI 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 ESXCLI, 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 ESXCLI commands directly against ESXi hosts in lockdown mode. See ESXCLI and Lockdown Mode and the *vSphere Security* documentation.

Order of Precedence for ESXCLI Host Management Commands

When you run a ESXCLI 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.

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.	*vSphere SDK for Perl Programming Guide*
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, 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.

You use the following options.

- `server` - Specifies the vCenter Server system that manages the host.
- `vihost` - Specifies the ESXi host.

Example

```
esxcli --server <vc_hostname_or_IP> --vihost <esxi_hostname_or_IP> --username <user_name> --
password <password> hardware clock get
```

Authenticating Directly to the Host

ESXCLI offers several options for authenticating directly to the host.

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 ESXCLI 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 Connection Options for ESXCLI Host Management Command Execution.

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

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

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

The following example illustrates how you can use the syntax.

```
esxcli --config <my_saved_config> network ip interface 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 file name 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.

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
<password> --vihost <esxi_hostname_or_IP> <namespace> [<namespace]...> <command> --
<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 <vc_hostname_or_IP> --username snow\-white --password dwarf\$ network ip
interface list
```

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

Example: Windows

```
esxcli --server <vc_hostname_or_IP> --username "snow-white" --password "dwarf$" network ip
interface list
```

Using the Microsoft Windows Security Support Provider Interface

With the `--passthroughauth` option, which is available if you run ESXCLI 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 ESXCLI commands and the vCenter Server software run on the same computer, the user needs only a local account to run the command. If the ESXCLI 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
```

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 with the `--list` option. The system does not prompt for a user name and password.

ESXCLI 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 Client or ESXCLI commands that support the `--vihost` option.

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

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 into a Web browser.
2. Click the **Download trusted root CA certificates** link.
3. Verify that the extension of the downloaded file is `.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` option 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` option 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 <vc_hostname_or_IP> --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
```

Connection Options for ESXCLI Host Management Command Execution

You can use connection options that are available for all ESXCLI host management commands.

The following table lists options that are available for all ESXCLI host management commands in alphabetical order. The table includes options for use on the command line and variables for use in configuration files.

Important ESXCLI 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>`
	Specifies 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.
	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.

Option and Environment Variable	Description
`--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. See the *vSphere SDK for Perl Programming Guide* for information on managing 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 ESXCLI to map to when it is run on a foreign language system.
`--passthroughauth` `VI_PASSTHROUGHAUTH`	If you specify this option, the system uses the Microsoft Windows Security Support Provider Interface (SSPI) for authentication. Trusted users are not prompted for a user name and password. See the Microsoft Web site for 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 ESXCLI 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.
`--savesessionfile <file>` `VI_SAVESESSIONFILE=<file>`	Saves a session to the specified file. The session expires if it is idle for 30 minutes.

Option and Environment Variable	Description
`--server <server>` `VI_SERVER=<server>`	Uses the specified ESXi or vCenter Server system. Default is `localhost`. 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, SHA-256, or SHA-512 host certificate thumbprint if no CA certificates file is provided in the `--cacertsfile` option. The thumbprint is returned by the server in the error message if you attempt to run a command without specifying a thumbprint or certificate file.
`--url <url>` `VI_URL=<url>`	Connects to the specified vSphere Web Services SDK URL.
`--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 an ESXCLI 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`.

Using ESXCLI Commands in Scripts

Most administrators run scripts to perform the same task repeatedly or to perform a task on multiple hosts. You can run ESXCLI 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 ESXCLI, you can test the installation by running a command from the Windows command prompt.

Procedure

1. Open a command prompt.
2. (Optional) Change to the ESXCLI installation directory.

   The default directory is `<system_drive>:\Program Files (x86)\VMware\esxcli`.
3. Run the command, including the connection options and other options.

   ```
   esxcli --server <vc_hostname_or_IP> --username <user_name> --password <password> network
   ip interface list
   ```

Run Host Management Commands from a Linux System

After installation, you can run ESXCLI commands from the command prompt.

Procedure

1. Open a command prompt.
2. (Optional) Change to the ESXCLI installation directory.

   The default directory is `/opt/vmware/esxcli`.
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.

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

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