ESXCLI Concepts and Examples

2 APR 2020
VMware vSphere 7.0
VMware ESXi 7.0
vCenter Server 7.0

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

ESXCLI Concepts and Examples explains how to use ESXCLI commands and includes command overviews and examples.

Intended Audience

This book is for experienced Windows or Linux system administrators who are familiar with vSphere administration tasks and data center operations and know how to use commands in scripts.

ESXCLI Command Overviews 1

This chapter provides an overview of ESXCLI, connection options, and discusses ESXCLI and lockdown mode.

This chapter includes the following topics:

Introduction to ESXCLI

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

You can install the ESXCLI command set on a supported Linux or Windows system. See *Getting Started with ESXCLI*. After installation, you can run ESXCLI commands from the Linux or Windows system. You can manage ESXi hosts with ESXCLI commands by specifying connection options such as the target host, user, and password or a configuration file. See Connection Options for ESXCLI Host Management Commands.

Targets and Protocols for ESXCLI Host Management Commands

Most ESXCLI commands are used to manage or retrieve information about one or more ESXi hosts. They can target an ESXi host or a vCenter Server system.

When you target a vCenter Server system, you can use `--vihost` to specify the ESXi host to run the command against. All commands support the HTTP and HTTPS protocols.

ESXCLI Commands Available on Different ESXi Hosts

The available ESXCLI commands depend on the ESXi host version.

When you run an ESXCLI command, you must know the commands supported on the target host. For example, if you run commands against ESXi 7.x hosts, ESXCLI 7.x commands are supported. If you run commands against ESXi 6.x hosts, ESXCLI 6.x commands are supported.

Some commands or command outputs are determined by the host type. In addition, VMware partners might develop custom ESXCLI commands that you can run on hosts where the partner VIB has been installed.

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

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` 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
```

Using ESXCLI Output

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

The `--formatter` option supports three values - `csv`, `xml`, and `keyvalue` and is used before any namespace.

The following example lists all file system information in CSV format.

```
esxcli --formatter=csv storage filesystem list
```

You can pipe the output to a file.

```
esxcli --formatter=keyvalue storage filesystem list > myfilesystemlist.txt
```

Important You should always use a formatter for consistent output.

Connection Options for ESXCLI Host Management Commands

You can run ESXCLI host management commands and other commands with several different connection options.

You can target hosts directly or target a vCenter Server system and specify the host you want to manage.

Important ESXCLI supports both the IPv4 protocol and the IPv6 protocol.

See the *Getting Started with ESXCLI* documentation for a complete list and examples.

ESXCLI Host Management Commands and Lockdown Mode

For additional security, an administrator can place one or more hosts managed by a vCenter Server system in lockdown mode. Lockdown mode affects login privileges for the ESXi host.

See the *vSphere Security* document for a detailed discussion of normal lockdown mode and strict lockdown mode, and of how to enable and disable them.

To make changes to ESXi systems in lockdown mode, you must go through a vCenter Server system that manages the ESXi system as the user `vpxuser` and include both the `--server` and `--vihost` options.

```
esxcli --server MyVC --vihost MyESXi storage filesystem list
```

The command prompts for the vCenter Server system user name and password.

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.

Managing Hosts

2

You can use host management commands to stop, reboot ESXi hosts, enter and exit maintenance mode, and manage modules. You can also manage components, base images, add-ons, and host updates.

vSphere 7.0 introduces components, base images, and add-ons. A component is an installation packaging element. One component might contain multiple VIBs. The component has a version number that is separate from the version numbers of any VIBs it contains, though it might be the same. Each VIB in a component can contain a device driver, a CIM module, or an application for communicating between the two. Components simplify the packaging and installation of installable items on ESXi. The base image is an ESXi image that VMware provides with every release of ESXi. The base image is a collection of components that is complete and can boot up a server. Base images have a user-readable name and a unique version that is updated with every major or minor release of ESXi. The add-on is a collection of components that does not represent a complete, bootable image. You cannot use vendor add-ons on their own. To customize an ESXi release, you must add a vendor add-on to an ESXi base image.

For information on updating ESXi hosts with the `esxcli software` command and on changing the host acceptance level to match the level of a VIB that you might want to use for an update, see the *VMware ESXi Upgrade* document.

This chapter includes the following topics:

Stopping and Rebooting Hosts with ESXCLI

You can shut down or reboot an ESXi host by using the vSphere Client or ESXCLI.

Shutting down a managed host disconnects it from the vCenter Server system, but does not remove the host from the inventory. You can shut down a single host or all hosts in a data center or cluster.

To shut down a host, run `esxcli system shutdown poweroff`. You must specify the `--reason` option and supply a reason for the shutdown. A `--delay` option allows you to specify a delay interval, in seconds.

To reboot a host, run `esxcli system shutdown reboot`. You must specify the `--reason` option and supply a reason for the reboot. A `--delay` option allows you to specify a delay interval, in seconds.

Enter and Exit Maintenance Mode with ESXCLI

You place a host in maintenance mode to service it, for example, to install more memory. A host enters or leaves maintenance mode only as the result of a user request.

`esxcli system maintenanceMode set` allows you to enable or disable maintenance mode. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 To enter maintenance mode, run the following command.

```
esxcli <conn_options> system maintenanceMode set --enable true
```

After all virtual machines on the host have been suspended or migrated, the host enters maintenance mode.

Note You cannot deploy or power on a virtual machine on hosts in maintenance mode.

2 To exit maintenance mode, run the following command.

```
esxcli <conn_options> system maintenanceMode set --enable false
```

Note If you attempt to exit maintenance mode when the host is no longer in maintenance mode, an error informs you that maintenance mode is already disabled.

Manage Modules with ESXCLI

The `esxcli system module` command supports setting and retrieving VMkernel module options. Not all VMkernel modules have settable module options.

The following example illustrates how to examine and enable a VMkernel module. Specify one of the connection options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 List information about the module.

```
esxcli <conn_options> system module list --module=<module_name>
```

The system returns the name, type, value, and description of the module.

2 (Optional) List all enabled or loaded modules.

```
esxcli <conn_options> system module list --enabled=true
esxcli <conn_options> system module list --loaded=true
```

3 Enable the module.

```
esxcli <conn_options> system module set --module=<module_name> --enabled=true
```

4 Set the parameter.

```
esxcli system module parameters set --module=<module_name> --parameter-string=<parameter_string>
```

5 Verify that the module is configured.

```
esxcli <conn_options> system module parameters list --module=<module_name>
```

Retrieve Information about Components, Base Image, and Add-On on a Host with ESXCLI

You can use ESXCLI to list the components on an ESXi host and retrieve detailed information about each component. You can also retrieve information about base image and add-on on an ESXi host.

The following example illustrates how to list all components, retrieve details about an individual component, and retrieve information about base image and add-on. Specify one of the connection options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 List all components installed on the host.

```
esxcli <conn_options> software component list
```

The list contains information about the name, display name, version, display version, vendor, creation date, and acceptance level of each component.

2 Retrieve detailed information about a specific component from the list by specifying the component name.

Note The component name can be either in the `<component_name>` or `<component_name>:<version>` format. If there are multiple components with the same name, you must specify the version. Component names are case-sensitive.

```
esxcli <conn_options> software component get –n <component_name>
```

3 Retrieve detailed information about the base image installed on the host.

```
esxcli <conn_options> software baseimage get
```

4 Retrieve detailed information about the add-on installed on the host.

```
esxcli <conn_options> software addon get
```

Manage Components with ESXCLI

You can use ESXCLI to retrieve information about components in a depot, install or update components, and remove components.

The following example illustrates how to list all components in a depot, retrieve details about an individual component in the depot, install or update a component on an ESXi host, and remove a component from the host. Installing a component update might be useful for driver troubleshooting purposes. You can remove any unnecessary components from the host. Specify one of the connection options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 List all components in the depot by specifying the full remote URL to the `index.xml` file of the online depot or the local absolute datastore path to an offline bundle `.zip` file.

```
esxcli <conn_options> software sources component list –d <depot_url_or_offline_bundle_path>
```

The list contains information about the name, display name, version, display version, vendor, creation date, and acceptance level of each component.

2 Retrieve detailed information about a specific component in the depot by specifying the component name.

Note The component name can be either in the `<component_name>` or `<component_name>:<version>` format. If there are multiple components with the same name, you must specify the version. Component names are case-sensitive.

```
esxcli <conn_options> software sources component get –n <component_name> –d
<depot_url_or_offline_bundle_path>
```

3 Install or update a component on the ESXi host by specifying the name and version of the component in the depot and the location of the depot.

```
esxcli <conn_options> software component apply –n <component_name>:<version> –d
<depot_url_or_offline_bundle_path>
```

4 Remove a component from the host.

Note The component name can be either in the <component_name> or <component_name>:<version> format. If there are multiple components with the same name, you must specify the version. Component names are case-sensitive.

```
esxcli <conn_options> software component remove –n <component_name>
```

Manage Base Images and Add-Ons with ESXCLI

You can use ESXCLI to retrieve information about base images and add-ons in a depot. You can also apply a complete image by using a JSON software specification that specifies the base image, add-on, and components to install on the ESXi host.

The following example illustrates how to list all base images and add-ons in a depot, retrieve details about a specific base image or add-on, and apply a complete image to a host. You can also verify the signatures of installed components after applying the complete image. Specify one of the connection options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 List all base images in the depot by specifying the full remote URL to the `index.xml` file of the online depot or the local absolute datastore path to an offline bundle `.zip` file.

```
esxcli <conn_options> software sources baseimage list –d <depot_url_or_offline_bundle_path>
```

The list contains information about the release ID, version, vendor, release date, and acceptance level of each base image.

2 Retrieve detailed information about a specific base image in the depot by specifying the base image version.

```
esxcli <conn_options> software sources baseimage get –b <base_image_version> –d
<depot_url_or_offline_bundle_path>
```

3 List all add-ons in the depot by specifying the full remote URL to the `index.xml` file of the online depot or the local absolute datastore path to an offline bundle `.zip` file.

```
esxcli <conn_options> software sources addon list –d <depot_url_or_offline_bundle_path>
```

The list contains information about the release ID, version, vendor, release date, and acceptance level of each add-on.

4 Retrieve detailed information about a specific add-on in the depot by specifying the add-on name.

Note The add-on name can be either in the <*add-on_name*> or <*add-on_name*>:<*version*> format. If there are multiple add-ons with the same name, you must specify the version. Add-on names are case-sensitive.

```
esxcli <conn_options> software sources addon get -a <add-on_name> -d
<depot_url_or_offline_bundle_path>
```

5 Prepare a JSON software specification.

- Obtain a JSON software specification exported from a vSphere Lifecycle Manager managed cluster.
- Create a custom JSON software specification.

  You can use the following JSON software specification syntax to apply a base image only.

  ```
  {
     "add_on": null,
     "base_image": {
         "version": "<base_image_version>"
     },
     "components": {}
  }
  ```

  You can use the following JSON software specification syntax to apply a base image, an add-on, and one or more components.

  ```
  {
     "base_image": {
         "version": "<base_image_version>"
     },
     "add_on": {
        "name": "<add-on_name>",
        "version": "<add-on_version>"
     },
     "components": {
        "<component_name>": "<component_version>"
     }
  }
  ```

6 Apply a complete image to the ESXi host by specifying the location of the JSON software specification and the location of the depot.

Note The location of the JSON software specification can be either a remote URL or a local file path. You can use a software specification exported from a vSphere Lifecycle Manager managed cluster. You can specify multiple depots.

```
esxcli <conn_options> software apply -s <location_of_software_spec>.json -d
<depot_url_or_offline_bundle_path>
```

7 Verify the signatures of components installed on the host.

Note If you have not rebooted the ESXi host after applying the complete image, you can verify the newly applied image by using the `--rebooting-image` option.

```
esxcli <conn_options> software component signature verify
```

A list of all installed components appears. The list contains the name, version, vendor, acceptance level, and signature verification result of each component.

Updating Hosts

When you add custom drivers or patches to a host, the process is called an update.

- Update ESXi 6.0 hosts with `esxcli software vib` commands discussed in the *vSphere Upgrade* documentation included in the vSphere 6.0 documentation set.
- Update ESXi 6.5 hosts with `esxcli software vib` commands discussed in the *vSphere Upgrade* documentation included in the vSphere 6.5 documentation set.
- Update ESXi 6.7 hosts with `esxcli software vib` commands discussed in the *VMware ESXi Upgrade* documentation included in the vSphere 6.7 documentation set.
- Update ESXi 7.0 hosts with `esxcli software vib` commands discussed in the *VMware ESXi Upgrade* documentation included in the vSphere 7.0 documentation set.

Managing Files

3

You can use ESXCLI to manage VMFS (Virtual Machine File System) volumes.

Note See Chapter 4 Managing Storage for information about storage manipulation commands.

This chapter includes the following topics:

Introduction to Virtual Machine File Management

You can use the vSphere Client or ESXCLI commands to access different types of storage devices that your ESXi host discovers and to deploy datastores on those devices.

Note Datastores are logical containers, analogous to file systems, that hide specifics of each storage device and provide a uniform model for storing virtual machine files. Datastores can be used for storing ISO images, virtual machine templates, and floppy images. The vSphere Client uses the term datastore exclusively. In ESXCLI, the term datastore, as well as VMFS or NFS volume, refer to the same logical container on the physical device.

Depending on the type of storage you use, datastores can be backed by the VMFS and NFS file system formats.

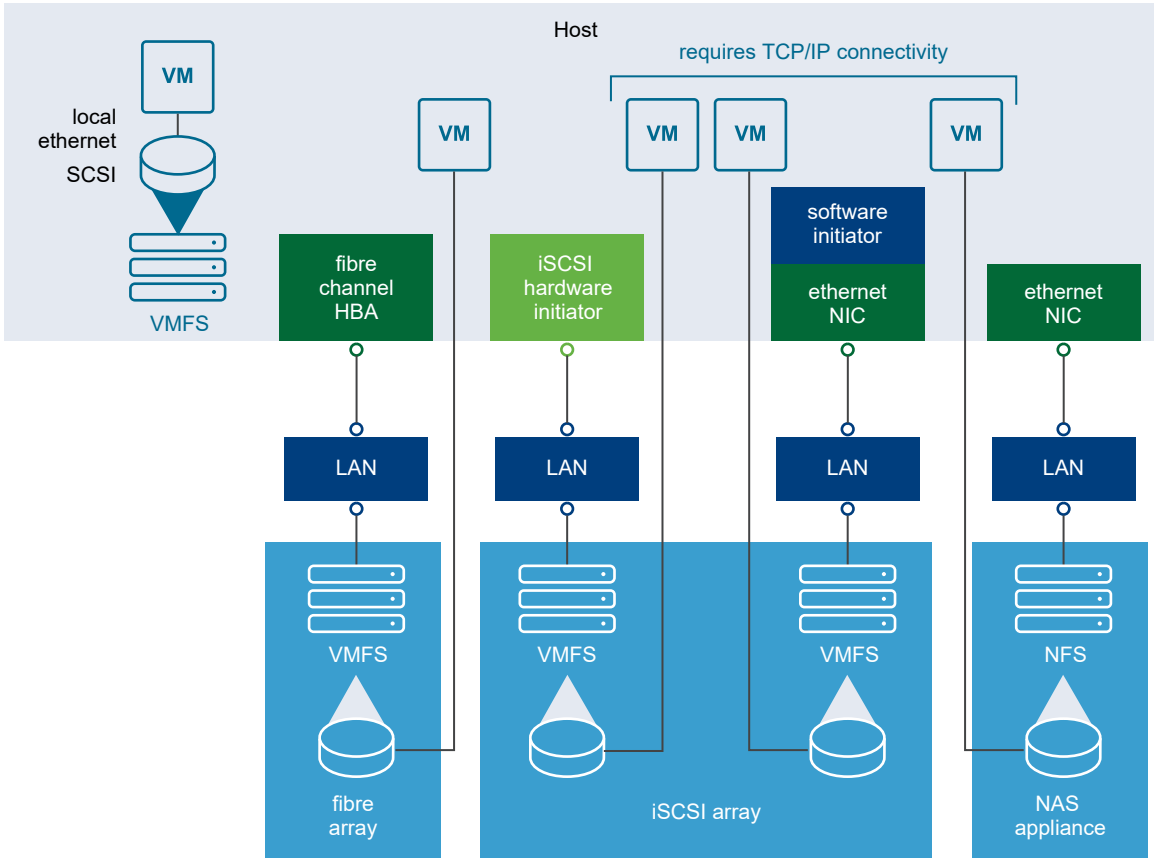
- Virtual Machine File System (VMFS) - High-performance file system that is optimized for storing virtual machines. Your host can deploy a VMFS datastore on any SCSI-based local or networked storage device, including Fibre Channel and iSCSI SAN equipment. As an alternative to using the VMFS datastore, your virtual machine can have direct access to raw devices and use a mapping file (RDM) as a proxy.

  You can manage VMFS and RDMs with the vSphere Client.

- Network File System (NFS) - The NFS client built into ESXi uses the NFS protocol over TCP/IP to access a designated NFS volume that is located on a NAS server. The ESXi host can mount the volume and use it for its storage needs. vSphere supports versions 3 and 4.1 of the NFS protocol. Typically, the NFS volume or directory is created by a storage administrator and is exported form the NFS server. The NFS volumes do not need to be formatted with a local file system, such as VMFS. You can mount the volumes directly and use them to store and boot virtual machines in the same way that you use VMFS datastores. The host can access a designated NFS volume located on an NFS server, mount the volume, and use it for any storage needs.

  You manage NAS storage devices from the vSphere Client or with the `esxcli storage nfs` command. The diagram below illustrates different types of storage, but it is for conceptual purposes only. It is not a recommended configuration.

Figure 3-1. Virtual Machines Accessing Different Types of Storage



Managing VMFS Volumes

Different commands are available for listing, mounting, and unmounting VMFS volumes and for listing, mounting, and unmounting VMFS snapshot volumes.

- Managing VMFS volumes

  `esxcli storage filesystem list` shows all volumes, mounted and unmounted, that are resolved, that is, that are not snapshot volumes.

`esxcli storage filesystem unmount` unmounts a currently mounted filesystem. Use this command for snapshot volumes or resolved volumes.

- Managing snapshot volumes

  `esxcli storage vmfs snapshot` commands can be used for listing, mounting, and resignaturing snapshot volumes. See Mounting Datastores with Existing Signatures and Resignaturing VMFS Copies.

Managing Duplicate VMFS Datastores

In some cases VMFS datastores can have duplicate UUIDs.

Each VMFS datastore created in a LUN has a unique UUID that is stored in the file system superblock. When the LUN is replicated or when a snapshot is made, the resulting LUN copy is identical, byte-for-byte, to the original LUN. As a result, if the original LUN contains a VMFS datastore with UUID X, the LUN copy appears to contain an identical VMFS datastore, or a VMFS datastore copy, with the same UUID X.

ESXi hosts can determine whether a LUN contains the VMFS datastore copy, and either mount the datastore copy with its original UUID or change the UUID to resignature the datastore.

When a LUN contains a VMFS datastore copy, you can mount the datastore with the existing signature or assign a new signature. The *vSphere Storage* documentation discusses volume resignaturing in detail.

Mounting Datastores with Existing Signatures

You can mount a VMFS datastore copy without changing its signature if the original is not mounted.

For example, you can maintain synchronized copies of virtual machines at a secondary site as part of a disaster recovery plan. In the event of a disaster at the primary site, you can mount the datastore copy and power on the virtual machines at the secondary site.

Important You can mount a VMFS datastore only if it does not conflict with an already mounted VMFS datastore that has the same UUID.

When you mount the VMFS datastore, ESXi allows both read and write operations to the datastore that resides on the LUN copy. The LUN copy must be writable. The datastore mounts are persistent and valid across system reboots.

Mount a Datastore with ESXCLI

The `esxcli storage filesystem` commands support mounting and unmounting volumes. You can also specify whether to persist the mounted volumes across reboots by using the `--no-persist` option.

Use the `esxcli storage filesystem` command to list mounted volumes, mount new volumes, and unmount a volume. Specify one of the connection options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 List all volumes that have been detected as snapshots.

```
esxcli <conn_options> storage filesystem list
```

2 Run `esxcli storage filesystem mount` with the volume label or volume UUID.

```
esxcli <conn_options> storage filesystem volume mount --volume-label=<label>|--volume-uuid=<VMFS-UUID>
```

Note This command fails if the original copy is online.

What to do next

You can later run `esxcli storage filesystem volume unmount` to unmount the snapshot volume.

```
esxcli <conn_options> storage filesystem volume unmount --volume-label=<label>|--volume-uuid=<VMFS-UUID>
```

Resignaturing VMFS Copies

You can use datastore resignaturing to retain the data stored on the VMFS datastore copy.

When resignaturing a VMFS copy, the ESXi host assigns a new UUID and a new label to the copy, and mounts the copy as a datastore distinct from the original. Because ESXi prevents you from resignaturing the mounted datastore, unmount the datastore before resignaturing.

The default format of the new label assigned to the datastore is `snap-<snapID>-<oldLabel>`, where `<snapID>` is an integer and `<oldLabel>` is the label of the original datastore.

When you perform datastore resignaturing, consider the following points.

- Datastore resignaturing is irreversible.
- The LUN copy that contains the VMFS datastore that you resignature is no longer treated as a LUN copy.
- A spanned datastore can be resignatured only if all its extents are online.
- The resignaturing process is crash and fault tolerant. If the process is interrupted, you can resume it later.
- You can mount the new VMFS datastore without a risk of its UUID conflicting with UUIDs of any other datastore, such as an ancestor or child in a hierarchy of LUN snapshots.

Resignature a VMFS Copy with ESXCLI

The `esxcli storage vmfs snapshot` commands support resignaturing a snapshot volume.

Specify one of the connection options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 List unresolved snapshots or replica volumes.

```
esxcli <conn_options> storage vmfs snapshot list
```

2 (Optional) Unmount the copy.

```
esxcli <conn_options> storage filesystem unmount
```

3 Run the resignature command.

```
esxcli <conn_options> storage vmfs snapshot resignature --volume-label=<label>|--volume-uuid=<id>
```

The command returns to the prompt or signals an error.

What to do next

After resignaturing, you might have to perform the following operations.

- If the resignatured datastore contains virtual machines, update references to the original VMFS datastore in the virtual machine files, including `.vmx`, `.vmdk`, `.vmsd`, and `.vmsn`.
- To power on virtual machines, register them with the vCenter Server system.

Reclaiming Unused Storage Space

When VMFS datastores reside on thin-provisioned LUNs, you can use ESXCLI commands to reclaim the unused logical blocks of a thin-provisioned LUN formatted with VMFS.

When you run the commands, you must specify the volume label `--volume-label` or the volume ID `--volume-uuid` but you cannot specify both.

In each iteration, the command issues unmap commands to the number of file system blocks that are specified by the optional `reclaim-unit` argument, which defaults to 200.

The following examples illustrate how to use the command.

```
# esxcli storage vmfs unmap --volume-label datastore1 --reclaim-unit 100
# esxcli storage vmfs unmap -l datastore1 -n 100
# esxcli storage vmfs unmap --volume-uuid 515615fb-1e65c01d-b40f-001d096dbf97 --reclaim-unit 500
# esxcli storage vmfs unmap -u 515615fb-1e65c01d-b40f-001d096dbf97 -n 500
```

```
# esxcli storage vmfs unmap -l datastore1
# esxcli storage vmfs unmap -u 515615fb-1e65c01d-b40f-001d096dbf97
```

Managing Storage

4

A virtual machine uses a virtual disk to store its operating system, program files, and other data associated with its activities. A virtual disk is a large physical file, or a set of files, that can be copied, moved, archived, and backed up.

To store virtual disk files and manipulate the files, a host requires dedicated storage space. ESXi storage is storage space on a variety of physical storage systems, local or networked, that a host uses to store virtual machine disks.

Chapter 5 Managing iSCSI Storage discusses iSCSI storage management. Chapter 6 Managing Third-Party Storage Arrays explains how to manage the Pluggable Storage Architecture, including Path Selection Plugin (PSP) and Storage Array Type Plug-in (SATP) configuration.

For information on masking and unmasking paths with ESXCLI, see the *vSphere Storage* documentation.

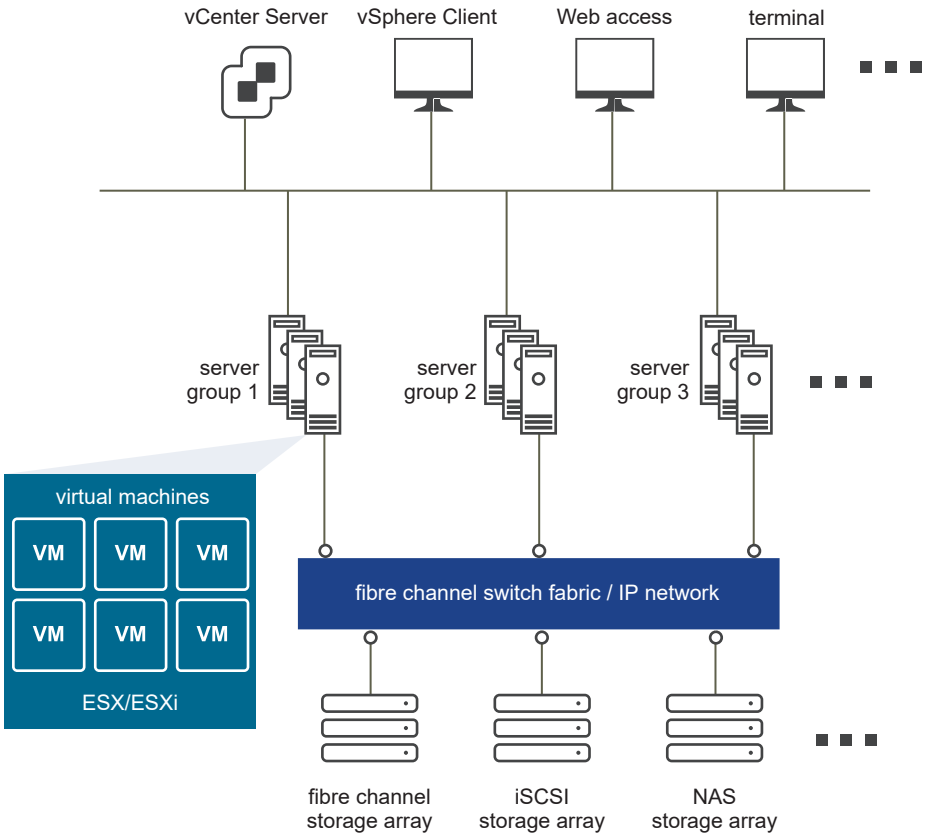
This chapter includes the following topics:

Introduction to Storage

Fibre Channel SAN arrays, iSCSI SAN arrays, and NAS arrays are widely used storage technologies supported by VMware vSphere to meet different data center storage needs.

The storage arrays are connected to and shared between groups of servers through storage area networks. This arrangement allows aggregation of the storage resources and provides more flexibility in provisioning them to virtual machines.

Figure 4-1. vSphere Data Center Physical Topology



How Virtual Machines Access Storage

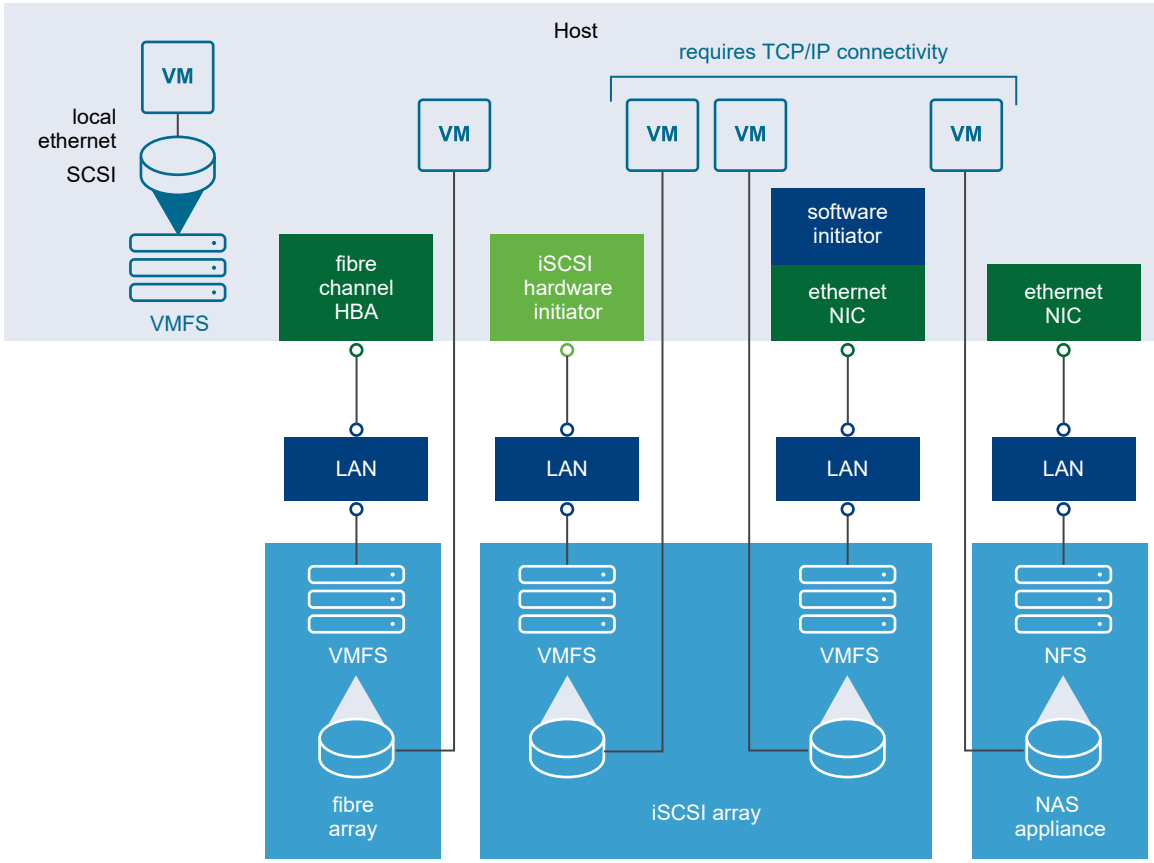
A virtual disk hides the physical storage layer from the virtual machine's operating system.

Regardless of the type of storage device that your host uses, the virtual disk always appears to the virtual machine as a mounted SCSI device. As a result, you can run operating systems that are not certified for specific storage equipment, such as SAN, in the virtual machine.

When a virtual machine communicates with its virtual disk stored on a datastore, it issues SCSI commands. Because datastores can exist on various types of physical storage, these commands are encapsulated into other forms, depending on the protocol that the ESXi host uses to connect to a storage device.

Figure 4-2. Virtual Machines Accessing Different Types of Storage depicts five virtual machines that use different types of storage to illustrate the differences between each type.

Figure 4-2. Virtual Machines Accessing Different Types of Storage



You can use ESXCLI commands to manage the virtual machine file system and storage devices.

- Datastores - Several commands allow you to manage datastores and are useful for multiple protocols.
  - LUNs - Use `esxcli storage core` to display available LUNs and mappings for each VMFS volume to its corresponding partition. See Examining LUNs.
  - Path management - Use `esxcli storage core` to list information about Fibre Channel or iSCSI LUNs and to change a path's state. See Managing Paths. Use the ESXCLI command to view and modify path policies. See Managing Path Policies.
  - Rescan - Use `esxcli storage core` to perform a rescan operation each time you reconfigure your storage setup. See Scanning Storage Adapters.

- Storage devices - Several commands manage only specific storage devices.
  - NFS storage - Use `esxcli storage nfs` to manage NAS storage devices. See Managing NFS/NAS Datastores.
  - iSCSI storage - Use `esxcli iscsi` to manage both hardware and software iSCSI. See Chapter 5 Managing iSCSI Storage.
- Software-defined storage - vSphere supports several types of software-defined storage.
  - vSAN storage - Use commands in the `esxcli vsan` namespace to manage vSAN. See Monitoring and Managing vSAN Storage.
  - Virtual Flash storage - Use commands in the `esxcli storage vflash` namespace to manage VMware vSphere Flash Read Cache.
  - Virtual volumes - Virtual volumes offer a different layer of abstraction than datastores. As a result, finer-grained management is possible. Use commands in the `esxcli storage vvol` namespace.

Datastores

ESXi hosts use storage space on a variety of physical storage systems, including internal and external devices and networked storage.

A host can discover storage devices to which it has access and format them as datastores. Each datastore is a special logical container, analogous to a file system on a logical volume, where the host places virtual disk files and other virtual machine files. Datastores hide specifics of each storage product and provide a uniform model for storing virtual machine files.

Depending on the type of storage you use, datastores can be backed by the following file system formats.

- Virtual Machine File System (VMFS) - High-performance file system optimized for storing virtual machines. Your host can deploy a VMFS datastore on any SCSI-based local or networked storage device, including Fibre Channel and iSCSI SAN equipment.
- Network File System (NFS) - File system on a NAS storage device. ESXi supports NFS version 3 and 4.1. The host can access a designated NFS volume located on an NFS server, mount the volume, and use it for any storage needs.

Storage Device Naming

Each storage device, or LUN, is identified by several device identifier names.

Device Identifiers

Depending on the type of storage, the ESXi host uses different algorithms and conventions to generate an identifier for each storage device.

- SCSI INQUIRY identifiers - The host uses the SCSI INQUIRY command to query a storage device and uses the resulting data, in particular the Page 83 information, to generate a unique identifier. SCSI INQUIRY device identifiers are unique across all hosts, persistent, and have one of the following formats.
  - `naa.<number>`
  - `t10.<number>`
  - `eui.<number>`

  These formats follow the T10 committee standards. See the SCSI-3 documentation on the T10 committe Web site for information on Page 83.
- Path-based identifier. If the device does not provide the information on Page 83 of the T10 committee SCSI-3 documentation, the host generates an `mpx.<path>` name, where `<path>` represents the first path to the device, for example, `mpx.vmhba1:C0:T1:L3`. This identifier can be used in the same way as the SCSI inquiry identifiers.

  The `mpx.` identifier is created for local devices on the assumption that their path names are unique. However, this identifier is neither unique nor persistent and could change after every boot.

  Typically, the path to the device has the following format.

  ```
  vmhba<adapter>:C<channel>:T<target>:L<LUN>
  ```

  - `vmbh<adapter>` is the name of the storage adapter. The name refers to the physical adapter on the host, not the SCSI controller used by the virtual machines.
  - `C<channel>` is the storage channel number. Software iSCSI adapters and dependent hardware adapters use the channel number to show multiple paths to the same target.
  - `T<target>` is the target number. Target numbering is determined by the host and might change if the mappings of targets that are visible to the host change. Targets that are shared by different hosts might not have the same target number.
  - `L<LUN>` is the LUN number that shows the position of the LUN within the target. The number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0).

Legacy Identifiers

In addition to the SCSI INQUIRY or mpx identifiers, ESXi generates an alternative legacy name, called VML name, for each device. Use the device UID instead.

Examining LUNs

A LUN (Logical Unit Number) is an identifier for a disk volume in a storage array target.

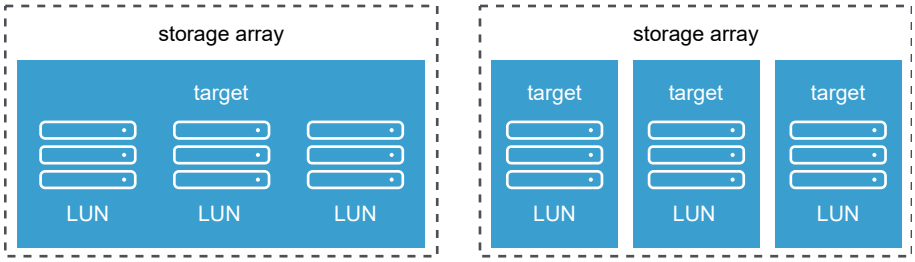
Target and Device Representation

In the ESXi context, the term target identifies a single storage unit that a host can access. The terms device and LUN describe a logical volume that represents storage space on a target.

The terms device and LUN mean a SCSI volume presented to the host from a storage target.

Different storage vendors present their storage systems to ESXi hosts in different ways. Some vendors present a single target with multiple LUNs on it. Other vendors, especially iSCSI vendors, present multiple targets with one LUN each.

Figure 4-3. Target and LUN Representations



In Figure 4-3. Target and LUN Representations, three LUNs are available in each configuration. On the left, the host sees one target, but that target has three LUNs that can be used. Each LUN represents an individual storage volume. On the right, the host sees three different targets, each having one LUN.

Examining LUNs with ESXCLI

You can use `esxcli storage core` to display information about available LUNs on ESXi.

You can run one of the following commands to examine LUNs. Specify one of the connection options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- List all logical devices known on this system with detailed information.

  ```
  esxcli <conn_options> storage core device list
  ```

  The command lists device information for all logical devices on this system. The information includes the name (UUID), device type, display name, and multipathing plugin. Specify the `--device` option to only list information about a specific device. See Storage Device Naming for background information.

  ```
  naa.5000c50037b3967e
  Display Name: <name> (naa.5000c50037b3967e)
  Has Settable Display Name: true
  Size: 953869
  Device Type: Direct-Access
  ...
  naa.500000e014e7a4e0
  Display Name: <name> (naa.500000e014e7a4e0)
  Has Settable Display Name: true
  Size: 70007
  Device Type: Direct-Access
  ```

```
...
mpx.vmhba0:C0:T0:L0
Display Name: Local <name> CD-ROM (mpx.vmhba0:C0:T0:L0)
Has Settable Display Name: false
Size: 0
Device Type: CD-ROM
```

- List a specific logical device with its detailed information.

  ```
  esxcli <conn_options> storage core device list -d mpx.vmhba32:C0:T1:L0
  ```

- List all device unique identifiers.

  ```
  esxcli <conn_options> storage core device list
  ```

  The command lists the primary UID for each device, such as `naa.xxx` or other primary name, and any other UIDs for each UID (VML name). You can specify `--device` to only list information for a specific device.

- Print mappings for VMFS volumes to the corresponding partition, path to that partition, VMFS UUID, extent number, and volume names.

  ```
  esxcli <conn_option> storage filesystem list
  ```

- Print HBA devices with identifying information.

  ```
  esxcli <conn_options> storage core adapter list
  ```

  The return value includes adapter and UID information.

- Print a mapping between HBAs and the devices it provides paths to.

  ```
  esxcli <conn_options> storage core path list
  ```

Detach a Device and Remove a LUN

Before you can remove a LUN, you must detach the corresponding device by using the vSphere Client, or the `esxcli storage core device set` command.

Detaching a device brings a device offline. Detaching a device does not impact path states. If the LUN is still visible, the path state is not set to dead.

Prerequisites

- Make sure you are familiar with virtual machine migration. See the *vCenter Server and Host Management* documentation.
- Make sure you are familiar with datastore mounting and unmounting. See Mount a Datastore with ESXCLI.

Procedure

1 Migrate virtual machines from the device you plan to detach.

2 Unmount the datastore deployed on the device.

If the unmount fails, ESXCLI returns an error. If you ignore that error, you will get an error when you attempt to detach a device with a VMFS partition still in use.

3 If the unmount failed, check whether the device is in use.

```
esxcli storage core device world list -d <device>
```

If a VMFS volume is using the device indirectly, the world name includes the string `idle0`. If a virtual machine uses the device as an RDM, the virtual machine process name is displayed. If any other process is using the raw device, the information is displayed.

4 Detach the storage device.

```
esxcli storage core device set -d naa.xxx... --state=off
```

Detach is persistent across reboots and device unregistration. Any device that is detached remains detached until a manual attach operation. Rescan does not bring persistently detached devices back online. A persistently detached device comes back in the off state.

ESXi maintains the persistent information about the device's offline state even if the device is unregistered. You can remove the device information by running `esxcli storage core device detached remove -d naa.12`.

5 (Optional) To troubleshoot the detach operation, list all devices that were detached manually.

```
esxcli storage core device detached list
```

6 Perform a rescan.

```
esxcli <conn_options> storage core adapter rescan
```

Reattach a Device

When you have completed storage reconfiguration, you can reattach the storage device, mount the datastore, and restart the virtual machines.

Prerequisites

Make sure you are familiar with datastore mounting. See Mounting Datastores with Existing Signatures.

Procedure

1 (Optional) Check whether the device is detached.

```
esxcli storage core device detached list
```

2 Attach the device.

```
esxcli storage core device set -d naa.XXX --state=on
```

3 Mount the datastore and restart virtual machines.

Working with Permanent Device Loss

In some cases a permanent device loss (PDL) might occur.

With earlier ESXi releases, an APD (All Paths Down) event results when the LUN becomes unavailable. The event is difficult for administrators because they do not have enough information about the state of the LUN to know which corrective action is appropriate.

The ESXi host can determine whether the cause of an APD event is temporary, or whether the cause is PDL. A PDL status occurs when the storage array returns SCSI sense codes indicating that the LUN is no longer available or that a severe, unrecoverable hardware problem exist with it. ESXi has an improved infrastructure that can speed up operations of upper-layer applications in a device loss scenario.

Important Do not plan for APD or PDL events, for example, when you want to upgrade your hardware. Instead, perform an orderly removal of LUNs from your ESXi server, which is described in Detach a Device and Remove a LUN, perform the operation, and add the LUN back.

Removing a PDL LUN

How you remove a PDL LUN depends on whether it was in use.

- If the LUN that goes into PDL is not in use by any user process or by the VMkernel, the LUN disappears by itself after a PDL.
- If the LUN was in use when it entered PLD, delete the LUN manually by following the process described in Detach a Device and Remove a LUN.

Reattach a PDL LUN

You can reattach a PDL LUN after it has been removed.

Procedure

1 Return the LUN to working order.

2 Remove any users of the device.

You cannot bring a device back without removing active users. The ESXi host cannot know whether the device that was added back has changed. ESXi must be able to treat the device similarly to a new device being discovered.

3 Perform a rescan to get the device back in working order.

Managing Paths

To maintain a constant connection between an ESXi host and its storage, ESXi supports multipathing. With multipathing you can use more than one physical path for transferring data between the ESXi host and the external storage device.

In case of failure of an element in the SAN network, such as an HBA, switch, or cable, the ESXi host can fail over to another physical path. On some devices, multipathing also offers load balancing, which redistributes I/O loads between multiple paths to reduce or eliminate potential bottlenecks.

The storage architecture supports a special VMkernel layer, Pluggable Storage Architecture (PSA). The PSA is an open modular framework that coordinates the simultaneous operation of multiple multipathing plug-ins (MPPs). You can manage PSA by using ESXCLI commands. See Chapter 6 Managing Third-Party Storage Arrays. This section assumes you are using only PSA plug-ins included in vSphere by default.

Multipathing with Local Storage and FC SANs

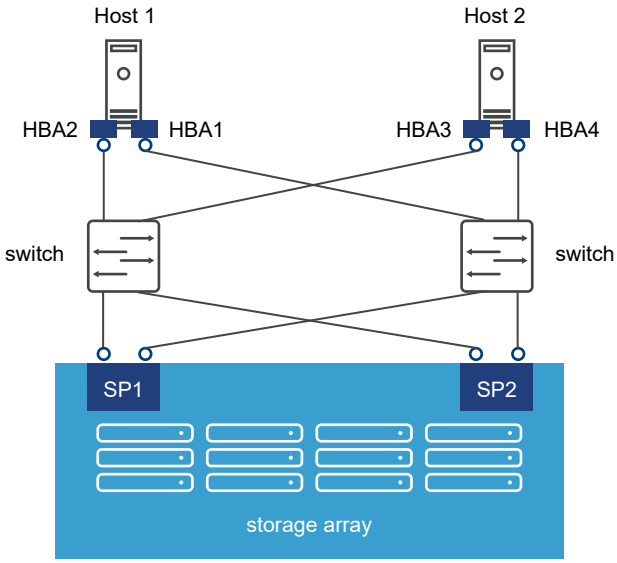
Multipathing is a technique that lets you use more than one physical path that transfers data between the host and an external storage device.

In a simple multipathing local storage topology, you can use one ESXi host with two HBAs. The ESXi host connects to a dual-port local storage system through two cables. This configuration ensures fault tolerance if one of the connection elements between the ESXi host and the local storage system fails.

To support path switching with FC SAN, the ESXi host typically has two HBAs available from which the storage array can be reached through one or more switches. Alternatively, the setup can include one HBA and two storage processors so that the HBA can use a different path to reach the disk array.

In FC Multipathing, multiple paths connect each host with the storage device. For example, if HBA1 or the link between HBA1 and the switch fails, HBA2 takes over and provides the connection between the server and the switch. The process of one HBA taking over for another is called HBA failover.

Figure 4-4. FC Multipathing



If SP1 or the link between SP1 and the switch breaks, SP2 takes over and provides the connection between the switch and the storage device. This process is called SP failover. ESXi multipathing supports HBA and SP failover.

After you have set up your hardware to support multipathing, you can use the vSphere Client or ESXCLI commands to list and manage paths. You can perform the following tasks.

- List path information. See Listing Path Information with ESXCLI.
- Change path state. See Disable a Path with ESXCLI.
- Change path policies. See Set Policy Details for Devices that Use Round Robin.
- Mask paths. See the *vSphere Storage* documentation.
- Manipulate the rules that match paths to multipathing plugins to newly discovered devices. See Managing Claim Rules.
- Run or rerun claim rules or unclaim paths. See Managing Claim Rules.
- Rescan storage adapters. See Scanning Storage Adapters.

Listing Path Information with ESXCLI

You can run `esxcli storage core path` to display information about Fibre Channel or iSCSI LUNs.

Important Use industry-standard device names, with format `eui.xxx` or `naa.xxx` to ensure consistency. Do not use VML LUN names unless device names are not available.

Names of virtual machine HBAs are not guaranteed to be valid across reboots.

You can display information about paths by running `esxcli storage core path`. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- List all devices with their corresponding paths, state of the path, adapter type, and other information.

  ```
  esxcli <conn_options> storage core path list
  ```

- Limit the display to only a specified path or device.

  ```
  esxcli <conn_options> storage core path list --path <path>
  esxcli <conn_options> storage core path list --device <device>
  ```

- List the statistics for the SCSI paths in the system. You can list all paths or limit the display to a specific path.

  ```
  esxcli <conn_options> storage core path stats get
  esxcli <conn_options> storage core path stats get --path <path>
  ```

- List detailed information for the paths for the device specified with `--device`.

  ```
  esxcli <conn_options> storage core path list -d <naa.xxxxxx>
  ```

- List all adapters.

  ```
  esxcli <conn_options> storage core adapter list
  ```

- Rescan all adapters.

  ```
  esxcli <conn_options> storage core adapter rescan
  ```

Disable a Path with ESXCLI

You can temporarily disable a path with ESXCLI for maintenance or other reasons, and enable the path when you need it again.

If you are changing a path's state, the change operation fails if I/O is active when the path setting is changed. Reissue the command. You must issue at least one I/O operation before the change takes effect.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 (Optional) List all devices and corresponding paths.

  ```
  esxcli <conn_options> storage core path list
  ```

  The display includes information about each path's state.

2 Set the state of a LUN path to off.

```
esxcli <conn_options> storage core path set --state off --path vmhba32:C0:T1:L0
```

What to do next

When you are ready, set the path state to active again.

```
esxcli <conn_options> storage core path set --state active --path vmhba32:C0:T1:L0
```

Managing Path Policies

For each storage device managed by NMP, and not PowerPath, an ESXi host uses a path selection policy. If you have a third-party PSP installed on your host, its policy also appears on the list.

Supported Path Policies

The following path policies are supported by default.

Policy	Description
VMW_PSP_FIXED	The host uses the designated preferred path, if it has been configured. Otherwise, the host selects the first working path discovered at system boot time. If you want the host to use a particular preferred path, specify it through the vSphere Client, or by using `esxcli storage nmp psp fixed deviceconfig set`. See Change the Path Policy with ESXCLI. The default policy for active-active storage devices is VMW_PSP_FIXED. **Note** If the host uses a default preferred path and the path's status turns to Dead, a new path is selected as preferred. However, if you explicitly designate the preferred path, it will remain preferred even when it becomes inaccessible.
VMW_PSP_MRU	The host selects the path that it used most recently. When the path becomes unavailable, the host selects an alternative path. The host does not revert back to the original path when that path becomes available again. There is no preferred path setting with the MRU policy. MRU is the default policy for active-passive storage devices. The VMW_PSP_MRU ranking capability allows you to assign ranks to individual paths. To set ranks to individual paths, use the `esxcli storage nmp psp generic pathconfig set` command. For details, see the VMware knowledge base article 2003468.
VMW_PSP_RR	The host uses an automatic path selection algorithm that rotates through all active paths when connecting to active-passive arrays, or through all available paths when connecting to active-active arrays. Automatic path selection implements load balancing across the physical paths available to your host. Load balancing is the process of spreading I/O requests across the paths. The goal is to optimize throughput performance such as I/O per second, megabytes per second, or response times. VMW_PSP_RR is the default for a number of arrays and can be used with both active-active and active-passive arrays to implement load balancing across paths for different LUNs.

Path Policy Effects

The type of array and the path policy determine the behavior of the host.

Policy	Active/Active Array	Active/Passive Array
Most Recently Used	Administrator action is required to fail back after path failure.	Administrator action is required to fail back after path failure.
Fixed	VMkernel resumes using the preferred path when connectivity is restored.	VMkernel attempts to resume by using the preferred path. This action can cause path thrashing or failure when another SP now owns the LUN.
Round Robin	No fail back.	Next path in round robin scheduling is selected.

Multipathing Considerations

You should consider a number of key points when working with multipathing.

The following considerations help you with multipathing.

- If no SATP is assigned to the device by the claim rules, the default SATP for iSCSI or FC devices is `VMW_SATP_DEFAULT_AA`. The default PSP is `VMW_PSP_FIXED`.
- When the system searches the SATP rules to locate a SATP for a given device, it searches the driver rules first. If there is no match, the vendor/model rules are searched, and finally the transport rules are searched. If no match occurs, NMP selects a default SATP for the device.
- If `VMW_SATP_ALUA` is assigned to a specific storage device, but the device is not ALUA-aware, no claim rule match occurs for this device. The device is claimed by the default SATP based on the device's transport type.
- The default PSP for all devices claimed by `VMW_SATP_ALUA` is `VMW_PSP_MRU`. The `VMW_PSP_MRU` selects an active/optimized path as reported by the `VMW_SATP_ALUA`, or an active/unoptimized path if there is no active/optimized path. This path is used until a better path is available (MRU). For example, if the `VMW_PSP_MRU` is currently using an active/unoptimized path and an active/optimized path becomes available, the `VMW_PSP_MRU` will switch the current path to the active/optimized one.
- While `VMW_PSP_MRU` is typically selected for ALUA arrays by default, certain ALUA storage arrays need to use `VMW_PSP_FIXED`. To check whether your storage array requires `VMW_PSP_FIXED`, see the *VMware Compatibility Guide* or contact your storage vendor. When using `VMW_PSP_FIXED` with ALUA arrays, unless you explicitly specify a preferred path, the ESXi host selects the most optimal working path and designates it as the default preferred path. If the host selected path becomes unavailable, the host selects an alternative available path. However, if you explicitly designate the preferred path, it remains preferred no matter what its status is.
- By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not delete this rule, unless you want to unmask these devices.

Change the Path Policy with ESXCLI

You can change the path policy with ESXCLI.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Prerequisites

Verify that you are familiar with the supported path policies. See Managing Path Policies.

Procedure

1 Ensure your device is claimed by the NMP plug-in.

Only NMP devices allow you to change the path policy.

```
esxcli <conn_options> storage nmp device list
```

2 Retrieve the list of path selection policies on the system to see which values are valid for the `--psp` option when you set the path policy.

```
esxcli storage core plugin registration list --plugin-class="PSP"
```

3 Set the path policy by using ESXCLI.

```
esxcli <conn_options> storage nmp device set --device naa.xxx --psp VMW_PSP_RR
```

4 (Optional) If you specified the `VMW_PSP_FIXED` policy, you must make sure the preferred path is set correctly.

a Check which path is the preferred path for a device.

```
esxcli <conn_options> storage nmp psp fixed deviceconfig get --device naa.xxx
```

b If necessary, change the preferred path.

```
esxcli <conn_options> storage nmp psp fixed deviceconfig set --device naa.xxx --path
vmhba3:C0:T5:L3
```

The command sets the preferred path to `vmhba3:C0:T5:L3`. Run the command with `--default` to clear the preferred path selection.

Set Policy Details for Devices that Use Round Robin

ESXi hosts can use multipathing for failover. With some storage devices, ESXi hosts can also use multipathing for load balancing.

To achieve better load balancing across paths, administrators can specify that the ESXi host should switch paths under specific circumstances. Different options determine when the ESXi host switches paths and what paths are chosen. Only a limited number of storage arrays support round robin.

You can use `esxcli storage nmp psp roundrobin` to retrieve and set round robin path options on a device controlled by the `roundrobin PSP`. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 Retrieve path selection settings for a device that is using the `roundrobin PSP`.

```
esxcli <conn_options> storage nmp psp roundrobin deviceconfig get --device na.xxx
```

2 Set the path selection. You can specify when the path should change, and whether unoptimized paths should be included.

- Use `--bytes` or `--iops` to specify when the path should change, as in the following examples.

  ```
  esxcli <conn_options> storage nmp psp roundrobin deviceconfig set --type "bytes" -B 12345 --
  device naa.xxx
  ```

  Sets the device specified by `--device` to switch to the next path each time 12345 bytes have been sent along the current path.

  ```
  esxcli <conn_options> storage nmp psp roundrobin deviceconfig set --type=iops --iops 4200 --
  device naa.xxx
  ```

  Sets the device specified by `--device` to switch after 4200 I/O operations have been performed on a path.
- Use `useano` to specify that the round robin PSP should include paths in the active, unoptimized state in the round robin set (1) or that the PSP should use active, unoptimized paths only if no active optimized paths are available (0). If you do not include this option, the PSP includes only active optimized paths in the round robin path set.

Scheduling Queues for Virtual Machine I/O

You can use ESXCLI to enable or disable per file I/O scheduling.

By default, vSphere provides a mechanism that creates scheduling queues for each virtual machine file. Each file has individual bandwidth controls. This mechanism ensures that the I/O for a particular virtual machine goes into its own separate queue and does not interfere with the I/O of other virtual machines.

This capability is enabled by default. You can turn it off by using the `esxcli system settings kernel set -s isPerFileSchedModelActive` option.

- Run `esxcli system settings kernel set -s isPerFileSchedModelActive -v FALSE` to disable per file scheduling.
- Run `esxcli system settings kernel set -s isPerFileSchedModelActive -v TRUE` to enable per file scheduling.

Managing NFS/NAS Datastores

ESXi hosts can access a designated NFS volume located on a NAS (Network Attached Storage) server, can mount the volume, and can use it for its storage needs. You can use NFS volumes to store and boot virtual machines in the same way that you use VMFS datastores.

Capabilities Supported by NFS/NAS

An NFS client built into the ESXi hypervisor uses the Network File System (NFS) protocol over TCP/IP to access a designated NFS volume that is located on a NAS server. The ESXi host can mount the volume and use it for its storage needs.

vSphere supports versions 3 and 4.1 of the NFS protocol.

Typically, the NFS volume or directory is created by a storage administrator and is exported from the NFS server. The NFS volume does not need to be formatted with a local file system, such as VMFS. You can mount the volume directly on ESXi hosts, and use it to store and boot virtual machines in the same way that you use VMFS datastores.

In addition to storing virtual disks on NFS datastores, you can also use NFS as a central repository for ISO images, virtual machine templates, and so on. If you use the datastore for ISO images, you can connect the virtual machine's CD-ROM device to an ISO file on the datastore and install a guest operating system from the ISO file.

ESXi hosts support the following shared storage capabilities on NFS volumes.

- VMware vMotion and Storage vMotion
- High Availability (HA), Fault Tolerance, and Distributed Resource Scheduler (DRS)
- ISO images, which are presented as CD-ROMs to virtual machines
- Virtual machine snapshots
- Host profiles
- Virtual machines with large capacity virtual disks, or disks greater than 2 TB. Virtual disks created on NFS datastores are thin-provisioned by default, unless you use hardware acceleration that supports the Reserve Space operation. See *Hardware Acceleration on NAS Devices* in the vSphere Storage documentation.

In addition to storing virtual disks on NFS datastores, you can also use NFS as a central repository for ISO images, virtual machine templates, and so on.

To use NFS as a shared repository, you create a directory on the NFS server and then mount the directory as a datastore on all hosts. If you use the datastore for ISO images, you can connect the virtual machine's CD-ROM device to an ISO file on the datastore and install a guest operating system from the ISO file.

Manage a NAS File System with ESXCLI

You can list, add, and delete a NAS file system with ESXCLI.

For more information on connection options, see Connection Options for ESXCLI Host Management Commands.

Procedure

1. List all known NAS file systems.

   ```
   esxcli <conn_options> storage nfs list
   ```

   For each NAS file system, the command lists the mount name, share name, and host name and whether the file system is mounted. If no NAS file systems are available, the system does not return a NAS filesystem and returns to the command prompt.

2. Add a new NAS file system to the ESXi host.

   Specify the NAS server with `--host`, the volume to use for the mount with `--volume-name`, and the share name on the remote system to use for this NAS mount point with `--share`.

   ```
   esxcli <conn_options> storage nfs add --host=dir42.eng.vmware.com --share=/<mount_dir> --volume-
   name=nfsstore-dir42
   ```

   This command adds an entry to the known NAS file system list and supplies the share name of the new NAS file system. You must supply the host name, share name, and volume name for the new NAS file system.

3. Add a second NAS file system with read-only access.

   ```
   esxcli <conn_options> storage nfs add --host=dir42.eng.vmware.com --share=/home --volume-
   name=FileServerHome2 --readonly
   ```

4. Delete one of the NAS file systems.

   ```
   esxcli <conn_options> storage nfs remove --volume-name=FileServerHome2
   ```

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

Monitor and Manage FibreChannel SAN Storage

The `esxcli storage san commands` help administrators troubleshoot issues with I/O devices and fabric, and include Fibre Channel, FCoE, iSCSI, SAS protocol statistics.

The commands allow you to retrieve device information and I/O statistics from those device. You can also issue Loop Initialization Primitives (LIP) to FC/FCoE devices and you can reset SAS devices.

For FC and FCoE devices, you can retrieve FC events such as `RSCN`, `LINKUP`, `LINKDOWN`, Frame Drop and `FCoE CVL`. The commands log a warning in the VMkernel log if it encounters too many Link Toggling or frame drops.

The following example examines and resets SAN storage through a FibreChannel adapter. Instead of fc, the information retrieval commands can also use `iscsi`, `fcoe`, and `sas`.

Procedure

1 List adapter attributes.

```
esxcli storage san fc list
```

2 Retrieve all events for a Fibre Channel I/O device.

```
esxcli storage san fc events get
```

3 Clear all I/O Device Management events for the specified adapter.

```
esxcli storage san fc events clear --adapter adapter
```

4 Reset the adapter.

```
esxcli storage san fc reset
```

Monitoring and Managing vSAN Storage

vSAN is a distributed layer of software that runs natively as a part of the ESXi hypervisor. vSAN aggregates local or direct-attached storage disks of a host cluster and creates a single storage pool shared across all hosts of the cluster.

While supporting VMware features that require shared storage, such as HA, vMotion, and DRS, vSAN eliminates the need for an external shared storage and simplifies storage configuration and virtual machine provisioning activities.

You can use ESXCLI commands to retrieve vSAN information, manage vSAN clusters, perform network management, add storage, set the policy, and perform other monitoring and management tasks. Type `esxcli vsan --help` for a complete list of commands.

Retrieve vSAN Information

You can use ESXCLI commands to retrieve vSAN information.

Procedure

1 Verify which VMkernel adapters are used for vSAN communication.

```
esxcli vsan network list
```

2 List storage disks that were claimed by vSAN.

```
esxcli vsan storage list
```

3 Get vSAN cluster information.

```
esxcli vsan cluster get
```

Manage a vSAN Cluster

You can activate vSAN when you create host clusters or enable vSAN on existing clusters. When enabled, vSAN aggregates all local storage disks available on the hosts into a single datastore shared by all hosts.

You can run these commands in the ESXi Shell for a host, or the command affects the target host that you specify as part of the ESXCLI connection options.

Procedure

1 Join the target host to a given vSAN cluster.

```
esxcli vsan cluster join --cluster-uuid <uuid>
```

Note The UUID of the cluster is required.

2 Verify that the target host is joined to a vSAN cluster.

```
esxcli vsan cluster get
```

3 Remove the target host from the vSAN cluster.

```
esxcli vsan cluster leave
```

Add and Remove vSAN Storage

You can use ESXCLI commands to add and remove vSAN storage. You can also retrieve information about the vSAN storage configuration.

Procedure

1 Add an HDD or data disk for use by vSAN.

```
esxcli vsan storage add --disks <device_name>
```

Note The command expects an empty disk, which is partitioned or formatted. Specify a device name, for example, `mpx.vmhba2:C0:T1:L0`.

2 Add an SSD disk for use by vSAN.

```
esxcli vsan storage add --ssd <device_name>
```

Note The command expects an empty disk, which is partitioned or formatted. Specify a device name, for example, `mpx.vmhba2:C0:T1:L0`.

3 List the vSAN storage configuration. You can display the complete list, or filter to show only a single device.

```
esxcli vsan storage list --device <device>
```

4 Remove disks or disk groups.

Note You can remove disks or disk groups only when vSAN is in manual mode. For the automatic disk claim mode, the remove action is not supported.

- Remove an individual vSAN disk.

```
esxcli vsan storage remove --disk <device_name>
```

  Instead of specifying the device name, you can specify the UUID if you include the `--uuid` option.

- Remove a disk group's SSD and each of its backing HDD drives from vSAN usage.

```
esxcli vsan storage remove --ssd <device_name>
```

  Instead of specifying the device name, you can specify the UUID if you include the `--uuid` option. Any SSD that you remove from vSAN becomes available for such features as Flash Read Cache.

Monitoring vSphere Flash Read Cache

Flash Read Cache™ lets you accelerate virtual machine performance through the use of host resident flash devices as a cache.

The *vSphere Storage* documentation discusses vSphere Flash Read Cache in some detail.

You can reserve a Flash Read Cache for any individual virtual disk. The Flash Read Cache is created only when a virtual machine is powered on, and it is discarded when a virtual machine is suspended or powered off. When you migrate a virtual machine you have the option to migrate the cache. By default the cache is migrated if the virtual flash module on the source and destination hosts are compatible. If you do not migrate the cache, the cache is rewarmed on the destination host. You can change the size of the cache while a virtual machine is powered on. In this instance, the existing cache is discarded and a new write-through cache is created, which results in a cache warm up period. The advantage of creating a new cache is that the cache size can better match the application's active data.

Flash Read Cache supports write-through or read caching. Write-back or write caching are not supported. Data reads are satisfied from the cache, if present. Data writes are dispatched to the backing storage, such as a SAN or NAS. All data that is read from or written to the backing storage is unconditionally stored in the cache.

Note Not all workloads benefit with a Flash Read Cache. The performance boost depends on your workload pattern and working set size. Read-intensive workloads with working sets that fit into the cache can benefit from a Flash Read Cache configuration. By configuring Flash Read Cache for your read-intensive workloads additional I/O resources become available on your shared storage, which can result in a performance increase for other workloads even though they are not configured to use Flash Read Cache.

You can manage vSphere Flash Read Cache from the vSphere Client. You can monitor Flash Read Cache by using commands in the `esxcli storage vflash` namespace. The following table lists available commands. See the *ESXCLI Reference* for a list of options to each command.

Table 4-1. Commands for Monitoring vSphere Flash Read Cache

Command	Description
`esxcli storage vflash cache get`	Gets individual `vflash` cache info.
`esxcli storage vflash cache list`	Lists individual `vflash` caches.
`esxcli storage vflash cache stats get`	Gets `vflash` cache statistics.
`esxcli storage vflash cache stats reset`	Resets `vflash` cache statistics.
`esxcli storage vflash device list`	Lists `vflash` SSD devices.
`esxcli storage vflash module get`	Gets `vflash` module info.
`esxcli storage vflash module list`	Lists `vflash` modules.
`esxcli storage vflash module stats get`	Gets `vflash` module statistics.

Monitoring and Managing Virtual Volumes

The Virtual Volumes functionality changes the storage management paradigm from managing space inside datastores to managing abstract storage objects handled by storage arrays.

With Virtual Volumes, an individual virtual machine, not the datastore, becomes a unit of storage management, while storage hardware gains complete control over virtual disk content, layout, and management. The *vSphere Storage* documentation discusses Virtual Volumes in some detail and explains how to manage them by using the vSphere Client.

The following ESXCLI commands are available for managing display information about virtual volumes and for unbinding all Virtual Volumes from all vendor providers. See the *vSphere Storage* documentation for information on creating Virtual Volumes and configuring multipathing and SCSI-based endpoints.

Table 4-2. VVol Commands

Command	Description
`esxcli storage vvol daemon unbindall`	Unbinds all Virtual Volume instances from all storage providers that are known to the ESXi host.
`esxcli storage vvol protocolendpoint list`	Lists the VVol protocol endpoints currently known to the ESXi host.
`esxcli storage vvol storagecontainer list`	Lists the VVol storage containers currently known to the ESXi host.
`esxcli storage vvol storagecontainer restore`	Restores storage containers of vendor providers that are registered on the host.
`esxcli storage vvol vasacontext get`	Gets the VASA context (VC UUID).
`esxcli storage vvol vendorprovider list`	Lists the vendor providers registered on the host.
`esxcli storage vvol vendorprovider restore`	Restores the vendor providers that are registered on the host.

Configuring FCoE Adapters

ESXi can use Fibre Channel over Ethernet (FCoE) adapters to access Fibre Channel storage.

The FCoE protocol encapsulates Fibre Channel frames into Ethernet frames. As a result, your host does not need special Fibre Channel links to connect to Fibre Channel storage, but can use 10 Gbit lossless Ethernet to deliver Fibre Channel traffic.

To use FCoE, you need to install FCoE adapters. The adapters that VMware supports generally fall into two categories, hardware FCoE adapters and software FCoE adapters.

- Hardware FCoE adapters include completely offloaded specialized Converged Network Adapters (CNAs) that contain network and Fibre Channel functionalities on the same card. When such an adapter is installed, your host detects and can use both CNA components. In the vSphere Client, the networking component appears as a standard network adapter (vmnic) and the Fibre Channel component as a FCoE adapter (vmhba). You do not have to configure a hardware FCoE adapter to be able to use it.
- A software FCoE adapter is a software code that performs some of the FCoE processing. The adapter can be used with a number of NICs that support partial FCoE offload. Unlike the hardware FCoE adapter, the software adapter must be activated.

Scanning Storage Adapters

You must perform a rescan operation each time you reconfigure your storage setup.

You can scan by using the vSphere Client or the `esxcli storage core adapter rescan` command.

`esxcli storage core adapter rescan` supports the following additional options.

- `-a|--all or -A|--adapter=<string>` – Scan all adapters or a specified adapter.
- `-S|--skip-claim` – Skip claiming of new devices by the appropriate multipath plug-in.
- `-F|--skip-fs-scan` – Skip filesystem scan.
- `-t|--type` – Specify the type of scan to perform. The command either scans for all changes (`all`) or for added, deleted, or updated adapters (`add, delete, update`).

The following command scans a specific adapter and skips the filesystem scan that is performed by default.

```
esxcli <conn_options> storage core adapter rescan --adapter=vmhba33 --skip-claim
```

The command returns an indication of success or failure, but no detailed information.

Retrieving SMART Information

You can use ESXCLI to retrieve information related to SMART. SMART is a monitoring system for computer hard disks that reports information about the disks.

You can use the following example syntax to retrieve SMART information.

```
esxcli storage core device smart get -d device
```

What the command returns depends on the level of SMART information that the device supports. If no information is available for a parameter, the output displays N/A, as in the following sample output.

```
Parameter                     Value  Threshold  Worst
-------------------------------------------------------
Health Status                 OK     N/A        N/A
Media Wearout Indicator       N/A    N/A        N/A
Write Error Count             N/A    N/A        N/A
Read Error Count              119    6          74
Power-on Hours                57     0          57
Power Cycle Count             100    20         100
Reallocated Sector Count      100    36         100
Raw Read Error Rate           119    6          74
Drive Temperature             38     0          49
Driver Rated Max Temperature  62     45         51
Write Sectors TOT Count       200    0          200
Read Sectors TOT Count        100    0          253
Initial Bad Block Count       N/A    N/A        N/A
```

Managing iSCSI Storage 5

ESXi systems include iSCSI technology to access remote storage using an IP network. You can use the vSphere Client or commands in the `esxcli iscsi` namespace to configure both hardware and software iSCSI storage for your ESXi system.

See the *vSphere Storage* documentation for additional information.

This chapter includes the following topics:

iSCSI Storage Overview

With iSCSI, SCSI storage commands that your virtual machine issues to its virtual disk are converted into TCP/IP protocol packets and transmitted to a remote device, or target, on which the virtual disk is located. To the virtual machine, the device appears as a locally attached SCSI drive.

To access remote targets, the ESXi host uses iSCSI initiators. Initiators transport SCSI requests and responses between ESXi and the target storage device on the IP network. ESXi supports the following types of initiators.

- Software iSCSI adapter - VMware code built into the VMkernel. Allows an ESXi host to connect to the iSCSI storage device through standard network adapters. The software initiator handles iSCSI processing while communicating with the network adapter.

- Hardware iSCSI adapter - Offloads all iSCSI and network processing from your host. Hardware iSCSI adapters are broken into two types.
  - Dependent hardware iSCSI adapter - Leverages the VMware iSCSI management and configuration interfaces.
  - Independent hardware iSCSI adapter - Leverages its own iSCSI management and configuration interfaces.

See the *vSphere Storage* documentation for details on setup and failover scenarios.

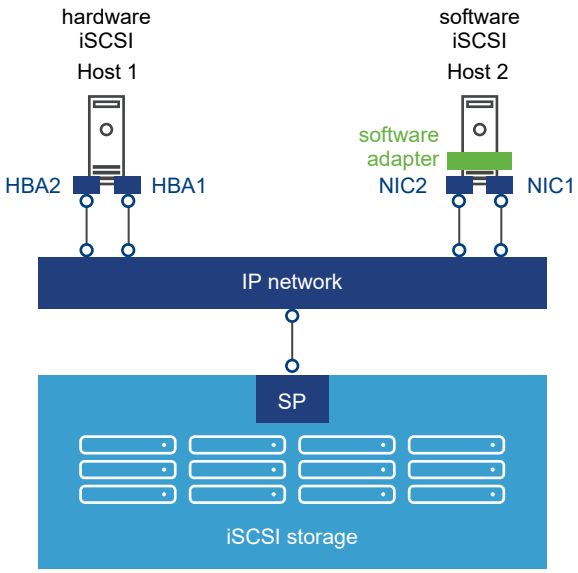
You must configure iSCSI initiators for the host to access and display iSCSI storage devices.

Figure 5-1. iSCSI Storage depicts hosts that use different types of iSCSI initiators.

- The host on the left uses an independent hardware iSCSI adapter to connect to the iSCSI storage system.
- The host on the right uses software iSCSI.

Dependent hardware iSCSI can be implemented in different ways and is not shown. iSCSI storage devices from the storage system become available to the host. You can access the storage devices and create VMFS datastores for your storage needs.

Figure 5-1. iSCSI Storage



Discovery Sessions

A discovery session is part of the iSCSI protocol. The discovery session returns the set of targets that you can access on an iSCSI storage system.

ESXi systems support dynamic and static discovery.

- Dynamic discovery - Also known as Send Targets discovery. Each time the ESXi host contacts a specified iSCSI storage server, it sends a Send Targets request to the server. In response, the iSCSI storage server supplies a list of available targets to the ESXi host. Monitor and manage with `esxcli iscsi adapter discovery sendtarget`.
- Static discovery - The ESXi host does not have to perform discovery. Instead, the ESXi host uses the IP addresses or domain names and iSCSI target names, IQN or EUI format names, to communicate with the iSCSI target. Monitor and manage with `esxcli iscsi adapter discovery statictarget`.

For either case, you set up target discovery addresses so that the initiator can determine which storage resource on the network is available for access. You can do this setup with dynamic discovery or static discovery. With dynamic discovery, all targets associated with an IP address or host name and the iSCSI name are discovered. With static discovery, you must specify the IP address or host name and the iSCSI name of the target you want to access. The iSCSI HBA must be in the same VLAN as both ports of the iSCSI array.

Discovery Target Names

The target name is either an IQN name or an EUI name.

The IQN and EUI names use specific formats.

- The IQN name uses the following format.

  ```
  iqn.yyyy-mm.{reversed domain name}:id_string
  ```

  The following IQN name contains example values.

  ```
  iqn.2007-05.com.mydomain:storage.tape.sys3.abc
  ```

  The ESXi host generates an IQN name for software iSCSI and dependent hardware iSCSI adapters. You can change that default IQN name.

- The EUI name is described in IETF rfc3720 as follows.

  The IEEE Registration Authority provides a service for assigning globally unique identifiers [EUI]. The EUI-64 format is used to build a global identifier in other network protocols. For example, Fibre Channel defines a method of encoding it into a `WorldWideName`.

  The format is `eui.` followed by an EUI-64 identifier (16 ASCII-encoded hexadecimal digits).

  The following EUI name contains example values.

  ```
  Type EUI-64 identifier (ASCII-encoded hexadecimal)
  +- -++---------------+
  |   ||               |
  eui.02004567A425678D
  ```

The IEEE EUI-64 iSCSI name format can be used when a manufacturer is registered with the IEEE Registration Authority and uses EUI-64 formatted worldwide unique names for its products.

You can check in the UI of the storage array whether an array uses an IQN name or an EUI name.

Protecting an iSCSI SAN

Your iSCSI configuration is only as secure as your IP network. By enforcing good security standards when you set up your network, you help safeguard your iSCSI storage.

Protecting Transmitted Data

A primary security risk in iSCSI SANs is that an attacker might sniff transmitted storage data.

Neither the iSCSI adapter nor the ESXi host iSCSI initiator encrypts the data that it transmits to and from the targets, making the data vulnerable to sniffing attacks. You must therefore take additional measures to prevent attackers from easily seeing iSCSI data.

Allowing your virtual machines to share virtual switches and VLANs with your iSCSI configuration potentially exposes iSCSI traffic to misuse by a virtual machine attacker. To help ensure that intruders cannot listen to iSCSI transmissions, make sure that none of your virtual machines can see the iSCSI storage network.

Protect your system by giving the iSCSI SAN a dedicated virtual switch.

- If you use an independent hardware iSCSI adapter, make sure that the iSCSI adapter and ESXi physical network adapter are not inadvertently connected outside the host. Such a connection might result from sharing a switch.
- If you use dependent hardware or software iscsi adapter, which uses ESXi networking, configure iSCSI storage through a different virtual switch than the one used by your virtual machines.

You can also configure your iSCSI SAN on its own VLAN to improve performance and security. Placing your iSCSI configuration on a separate VLAN ensures that no devices other than the iSCSI adapter can see transmissions within the iSCSI SAN. With a dedicated VLAN, network congestion from other sources cannot interfere with iSCSI traffic.

Securing iSCSI Ports

You can improve the security of iSCSI ports by installing security patches and limiting the devices connected to the iSCSI network.

When you run iSCSI devices, the ESXi host does not open ports that listen for network connections. This measure reduces the chances that an intruder can break into the ESXi host through spare ports and gain control over the host. Therefore, running iSCSI does not present an additional security risks at the ESXi host end of the connection.

An iSCSI target device must have one or more open TCP ports to listen for iSCSI connections. If security vulnerabilities exist in the iSCSI device software, your data can be at risk through no fault of the ESXi system. To lower this risk, install all security patches that your storage equipment manufacturer provides and limit the devices connected to the iSCSI network.

Setting iSCSI CHAP

iSCSI storage systems authenticate an initiator using a name and key pair. ESXi systems support Challenge Handshake Authentication Protocol (CHAP).

Using CHAP for your SAN implementation is a best practice. The ESXi host and the iSCSI storage system must have CHAP enabled and must have common credentials. During iSCSI login, the iSCSI storage system exchanges its credentials with the ESXi system and checks them.

You can set up iSCSI authentication by using the vSphere Client, as discussed in the *vSphere Storage* documentation or by using the `esxcli` command, discussed in Enabling iSCSI Authentication. To use CHAP authentication, you must enable CHAP on both the initiator side and the storage system side. After authentication is enabled, it applies for targets to which no connection has been established, but does not apply to targets to which a connection is established. After the discovery address is set, the new volumes to which you add a connection are exposed and can be used.

For software iSCSI and dependent hardware iSCSI, ESXi hosts support per-discovery and per-target CHAP credentials. For independent hardware iSCSI, ESXi hosts support only one set of CHAP credentials per initiator. You cannot assign different CHAP credentials for different targets.

When you configure independent hardware iSCSI initiators, ensure that the CHAP configuration matches your iSCSI storage. If CHAP is enabled on the storage array, it must be enabled on the initiator. If CHAP is enabled, you must set up the CHAP authentication credentials on the ESXi host to match the credentials on the iSCSI storage.

Supported CHAP Levels

To set CHAP levels with `esxcli iscsi adapter setauth`, specify one of the values in Table 5-1. Supported Levels for CHAP for `<level>`. Only two levels are supported for independent hardware iSCSI.

Mutual CHAP is supported for software iSCSI and for dependent hardware iSCSI, but not for independent hardware iSCSI.

Important Ensure that CHAP is set to `chapRequired` before you set mutual CHAP, and use compatible levels for CHAP and mutual CHAP. Use different passwords for CHAP and mutual CHAP to avoid security risks.

Table 5-1. Supported Levels for CHAP

Level	Description	Supported
`chapProhibited`	Host does not use CHAP authentication. If authentication is enabled, specify `chapProhibited` to disable it.	Software iSCSI Dependent hardware iSCSI Independent hardware iSCSI
`chapDiscouraged`	Host uses a non-CHAP connection, but allows a CHAP connection as fallback.	Software iSCSI Dependent hardware iSCSI

Table 5-1. Supported Levels for CHAP (continued)

Level	Description	Supported
`chapPreferred`	Host uses CHAP if the CHAP connection succeeds, but uses non-CHAP connections as fallback.	Software iSCSI Dependent hardware iSCSI Independent hardware iSCSI
`chapRequired`	Host requires successful CHAP authentication. The connection fails if CHAP negotiation fails.	Software iSCSI Dependent hardware iSCSI

Returning Authentication to Default Inheritance

The values of iSCSI authentication settings associated with a dynamic discovery address or a static discovery target are inherited from the corresponding settings of the parent. For the dynamic discovery address, the parent is the adapter. For the static target, the parent is the adapter or discovery address.

- If you use the vSphere Client to modify authentication settings, you must deselect the **Inherit from Parent** check box before you can make a change to the discovery address or discovery target.
- If you use `esxcli iscsi` commands, the value you set overrides the inherited value. You can set CHAP at the following levels.
  - `esxcli iscsi adapter auth chap [get|set]`
  - `esxcli iscsi adapter discovery sendtarget auth chap [get|set]`
  - `esxcli iscsi adapter target portal auth chap [get|set]`

Inheritance is relevant only if you want to return a dynamic discovery address or a static discovery target to its inherited value. In that case, use one of the following commands.

- Dynamic discovery

  ```
  esxcli iscsi adapter discovery sendtarget auth chap set --inherit
  ```

- Static discovery

  ```
  esxcli iscsi adapter target portal auth chap set --inherit
  ```

Note You can set target-level CHAP authentication properties to be inherited from the send target level and set send target level CHAP authentication properties to be inherited from the adapter level. Resetting adapter-level properties is not supported.

Command Syntax for esxcli iscsi

You can manage iSCSI storage by using `esxcli iscsi` commands.

For details, see the *ESXCLI Reference* and esxcli iscsi Command Syntax.

esxcli iscsi Command Syntax

The `esxcli iscsi` command includes a number of nested namespaces.

The following table illustrates the namespace hierarchy. Commands at each level are included in bold. Many namespaces include both commands and namespaces.

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

Key to esxcli iscsi Short Options

ESXCLI commands for iSCSI management consistently use the same short options. For several options, the associated full option depends on the command.

Table 5-3. Short Options for iSCSI ESXCLI Command Options

Lower-case Option	Option	Upper-case Option	Option	Number	Option
a	--address, alias	A	--adapter	1	--dns1
c	--cid			2	--dns2
d	--direction	D	--default		

Table 5-3. Short Options for iSCSI ESXCLI Command Options (continued)

Lower-case Option	Option	Upper-case Option	Option	Number	Option
f	--file, force				
g	--gateway				
i	--ip	I	--inherit		
k	--key				
l	--level				
m	--method	M	--module		
n	--nic	N	--authname, --name		
o	--option				
p	--plugin				
s	--isid, subnet, switch	S	--state, secret		
v	--value				

iSCSI Storage Setup with ESXCLI

You can set up iSCSI storage by using commands in the `esxcli iscsi` namespace.

You can also set up iSCSI storage by using the vSphere Client.

Set Up Software iSCSI with ESXCLI

Software iSCSI setup requires a number of high-level tasks.

You should be familiar with the corresponding command for each task. You can refer to the relevant documentation for each command or run `esxcli iscsi --help` in the console. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Prerequisites

- Verify that you are familiar with iSCSI authentication. See Enabling iSCSI Authentication.
- Verify that you are familiar with CHAP. See Setting iSCSI CHAP.
- Verify that you are familiar with iSCSI parameters. See Listing and Setting iSCSI Parameters.

Procedure

1 Enable software iSCSI.

```
esxcli <conn_options> iscsi software set --enabled=true
```

2 Check whether a network portal, that is, a bound port, exists for iSCSI traffic.

```
esxcli <conn_options> iscsi adapter list
```

3 If no adapter exists, add one.

Software iSCSI does not require port binding, but requires that at least one VMkernel NIC is available and can be used as an iSCSI NIC. You can name the adapter as you add it.

```
esxcli <conn_options> iscsi networkportal add –n <portal_name> –A <vmhba>
```

4 (Optional) Check the status.

```
esxcli <conn_options> iscsi software get
```

The system prints `true` if software iSCSI is enabled, or `false` if it is not enabled.

5 (Optional) Set the iSCSI name and alias.

```
esxcli <conn_options> iscsi adapter set ––adapter=<iscsi adapter> ––name=<name>
esxcli <conn_options> iscsi adapter set ––adapter=<iscsi adapter> ––alias=<alias>
```

6 Add a dynamic discovery address or a static discovery address.

- With dynamic discovery, all storage targets associated with a host name or IP address are discovered. You can run the following command.

  ```
  esxcli <conn_options> iscsi adapter discovery sendtarget add ––address=<ip/dns[:port]> ––
  adapter=<adapter_name>
  ```

- With static discovery, you must specify the host name or IP address and the iSCSI name of the storage target. You can run the following command.

  ```
  esxcli <conn_options> iscsi adapter discovery statictarget add ––address=<ip/dns[:port]> ––
  adapter=<adapter_name> ––name=<target_name>
  ```

When you later remove a discovery address, it might still be displayed as the parent of a static target. You can add the discovery address and rescan to display the correct parent for the static targets.

7 (Optional) Set the authentication information for CHAP.

You can set per-target CHAP for static targets, per-adapter CHAP, or apply the command to the discovery address.

Option	Command
Adapter-level CHAP	`esxcli iscsi adapter auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba>`
Discovery-level CHAP	`esxcli iscsi adapter discovery sendtarget auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba> --address<sendtarget_address>`
Target-level CHAP	`esxcli iscsi adapter target portal auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba> --name<iscsi_iqn_name>`

The following example sets adapter-level CHAP.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=preferred --secret=uni_secret --adapter=vmhba33
```

8 (Optional) Set the authentication information for mutual CHAP by running `esxcli iscsi adapter auth chap set` again with `--direction` set to `mutual` and a different authentication user name and secret.

Option	Command
Adapter-level CHAP	`esxcli iscsi adapter auth chap set --direction=mutual --mchap_username=<name2> --mchap_password=<pwd2> --level=[prohibited required] --secret=<string2> --adapter=<vmhba>`
Discovery-level CHAP	`esxcli iscsi adapter discovery sendtarget auth chap set --direction=mutual --mchap_username=<name2> --mchap_password=<pwd2> --level=[prohibited, required] --secret=<string2> --adapter=<vmhba> --address=<sendtarget_address>`
Target-level CHAP	`esxcli iscsi adapter target portal auth chap set --direction=mutual --mchap_username=<nam2e> --mchap_password=<pwd2> --level=[prohibited required] --secret=<string2> --adapter=<vmhba> --name=<iscsi_iqn_name>`

Important You are responsible for making sure that CHAP is set before you set mutual CHAP, and for using compatible levels for CHAP and mutual CHAP.

9 (Optional) Set iSCSI parameters.

Option	Command
Adapter-level CHAP	`esxcli iscsi adapter param set --adapter=<vmhba> --key=<key> --value=<value>`
Discovery-level CHAP	`esxcli iscsi adapter discovery sendtarget param set --adapter=<vmhba> --key=<key> --value=<value> --address=<sendtarget_address>`
Target-level CHAP	`esxcli iscsi adapter target portal param set --adapter=<vmhba> --key=<key> --value=<value> --address=<address> --name=<iqn.name>`

10 After setup is complete, perform rediscovery and rescan all storage devices.

The following example performs the rediscovery and rescan operations.

```
esxcli <conn_options> iscsi adapter discovery rediscover
esxcli <conn_options> storage core adapter rescan --adapter=vmhba36
```

11 (Optional) If you want to make additional iSCSI login parameter changes, you must log out of the corresponding iSCSI session and log back in.

a Run `esxcli iscsi session remove` to log out.

b Run `esxcli iscsi session add` or rescan the adapter to add the session back.

Set Up Dependent Hardware iSCSI with ESXCLI

Dependent hardware iSCSI setup requires several high-level tasks.

You should be familiar with the corresponding command for each task. You can refer to the relevant documentation for each command or run `esxcli iscsi --help` in the console. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Prerequisites

- Verify that you are familiar with iSCSI authentication. See Enabling iSCSI Authentication.
- Verify that you are familiar with CHAP. See Setting iSCSI CHAP.
- Verify that you are familiar with iSCSI parameters. See Listing and Setting iSCSI Parameters.

Procedure

1 Determine the iSCSI adapter type and retrieve the iSCSI adapter ID.

```
esxcli <conn_options> iscsi adapter list
```

2 (Optional) Set the iSCSI name and alias.

```
esxcli <conn_options> iscsi adapter set --adapter <adapter_name> --name=<name>
esxcli <conn_options> iscsi adapter set --adapter <adapter_name> --alias=<alias>
```

3 Set up port binding.

a Identify the VMkernel port of the dependent hardware iSCSI adapter.

```
esxcli <conn_options> iscsi logicalnetworkportal list --adapter=<adapter_name>
```

b Connect the dependent hardware iSCSI initiator to the iSCSI VMkernel ports by running the following command for each port.

```
esxcli <conn_options> iscsi networkportal add --nic=<bound_vmknic> --adapter=<iscsi_adapter>
```

c Verify that the ports were added to the dependent hardware iSCSI initiator.

```
esxcli <conn_options> iscsi physicalnetworkportal list --adapter=<adapter_name>
```

4 Add a dynamic discovery address or a static discovery address.

- With dynamic discovery, all storage targets associated with a host name or IP address are discovered. You can run the following command.

```
esxcli <conn_options> iscsi adapter discovery sendtarget add --address=<ip/dns[:port]> --
adapter=<adapter_name>
```

- With static discovery, you must specify the host name or IP address and the iSCSI name of the storage target. You can run the following command.

```
esxcli <conn_options> iscsi adapter discovery statictarget add --address=<ip/dns[:port]> --
adapter=<adapter_name> --name=<target_name>
```

When you later remove a discovery address, it might still be displayed as the parent of a static target. You can add the discovery address and rescan to display the correct parent for the static targets.

5 (Optional) Set the authentication information for CHAP.

You can set per-target CHAP for static targets, per-adapter CHAP, or apply the command to the discovery address.

Option	Command
Adapter-level CHAP	`esxcli iscsi adapter auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba>`
Discovery-level CHAP	`esxcli iscsi adapter discovery sendtarget auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba> --address<sendtarget_address>`
Target-level CHAP	`esxcli iscsi adapter target portal auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba> --name<iscsi_iqn_name>`

The following example sets adapter-level CHAP.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=preferred --secret=uni_secret --adapter=vmhba33
```

6 (Optional) Set the authentication information for mutual CHAP by running `esxcli iscsi adapter auth chap set` again with `--direction` set to `mutual` and a different authentication user name and secret.

Option	Command
Adapter-level CHAP	`esxcli iscsi adapter auth chap set --direction=mutual --mchap_username=<name2> --mchap_password=<pwd2> --level=[prohibited required] --secret=<string2> --adapter=<vmhba>`
Discovery-level CHAP	`esxcli iscsi adapter discovery sendtarget auth chap set --direction=mutual --mchap_username=<name2> --mchap_password=<pwd2> --level=[prohibited, required] --secret=<string2> --adapter=<vmhba> --address=<sendtarget_address>`
Target-level CHAP	`esxcli iscsi adapter target portal auth chap set --direction=mutual --mchap_username=<nam2e> --mchap_password=<pwd2> --level=[prohibited required] --secret=<string2> --adapter=<vmhba> --name=<iscsi_iqn_name>`

Important You are responsible for making sure that CHAP is set before you set mutual CHAP, and for using compatible levels for CHAP and mutual CHAP.

7 (Optional) Set iSCSI parameters.

Option	Command
Adapter-level CHAP	`esxcli iscsi adapter param set --adapter=<vmhba> --key=<key> --value=<value>`
Discovery-level CHAP	`esxcli iscsi adapter discovery sendtarget param set --adapter=<vmhba> --key=<key> --value=<value> --address=<sendtarget_address>`
Target-level CHAP	`esxcli iscsi adapter target portal param set --adapter=<vmhba> --key=<key> --value=<value> --address=<address> --name=<iqn.name>`

8 After setup is complete, perform rediscovery and rescan all storage devices.

The following example performs the rediscovery and rescan operations.

```
esxcli <conn_options> iscsi adapter discovery rediscover
esxcli <conn_options> storage core adapter rescan --adapter=vmhba36
```

9 (Optional) If you want to make additional iSCSI login parameter changes, you must log out of the corresponding iSCSI session and log back in.

a Run `esxcli iscsi session remove` to log out.

b Run `esxcli iscsi session add` or rescan the adapter to add the session back.

Set Up Independent Hardware iSCSI with ESXCLI

With independent hardware-based iSCSI storage, you use a specialized third-party adapter capable of accessing iSCSI storage over TCP/IP. This iSCSI initiator handles all iSCSI and network processing and management for your ESXi system.

You must install and configure the independent hardware iSCSI adapter for your host before you can access the iSCSI storage device. For installation information, see vendor documentation.

Hardware iSCSI setup requires a number of high-level tasks. You should be familiar with the corresponding command for each task. You can refer to the relevant documentation for each command or run `esxcli iscsi --help` in the console. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Prerequisites

- Verify that you are familiar with iSCSI authentication. See Enabling iSCSI Authentication.
- Verify that you are familiar with CHAP. See Setting iSCSI CHAP.
- Verify that you are familiar with iSCSI parameters. See Listing and Setting iSCSI Parameters.

Procedure

1 Determine the iSCSI adapter type and retrieve the iSCSI adapter ID.

```
esxcli <conn_options> iscsi adapter list
```

2 Configure the hardware initiator (HBA) by running `esxcli iscsi networkportal ipconfig` with one or more of the following options.

Option	Description
`-A|--adapter=<str>`	iSCSI adapter name (required)
`-1|--dns1=<str>`	iSCSI network portal primary DNS address
`-2|--dns2=<str>`	iSCSI network portal secondary DNS address
`-g|--gateway=<str>`	iSCSI network portal gateway address
`-i|--ip=<str>`	iSCSI network portal IP address (required)
`-n|--nic=<str>`	iSCSI network portal (vmknic)
`-s|--subnet=<str>`	iSCSI network portal subnet mask (required)

3 (Optional) Set the iSCSI name and alias.

```
esxcli <conn_options> iscsi adapter set --adapter <adapter_name> --name=<name>
esxcli <conn_options> iscsi adapter set --adapter <adapter_name> --alias=<alias>
```

4 Add a dynamic discovery address or a static discovery address.

- With dynamic discovery, all storage targets associated with a host name or IP address are discovered. You can run the following command.

  ```
  esxcli <conn_options> iscsi adapter discovery sendtarget add --address=<ip/dns[:port]> --
  adapter=<adapter_name>
  ```

- With static discovery, you must specify the host name or IP address and the iSCSI name of the storage target. You can run the following command.

  ```
  esxcli <conn_options> iscsi adapter discovery statictarget add --address=<ip/dns[:port]> --
  adapter=<adapter_name> --name=<target_name>
  ```

5 (Optional) Set the authentication information for CHAP.

You can set per-target CHAP for static targets, per-adapter CHAP, or apply the command to the discovery address.

Option	Command
Adapter-level CHAP	`esxcli iscsi adapter auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba>`
Discovery-level CHAP	`esxcli iscsi adapter discovery sendtarget auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba> --address<sendtarget_address>`
Target-level CHAP	`esxcli iscsi adapter target portal auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --adapter=<vmhba> --name<iscsi_iqn_name>`

The following example sets adapter-level CHAP.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name> --chap_password=<pwd> --level=preferred --secret=uni_secret --adapter=vmhba33
```

Note Mutual CHAP is not supported for independent hardware iSCSI storage.

6 (Optional) Set iSCSI parameters.

Option	Command
Adapter-level CHAP	`esxcli iscsi adapter param set --adapter=<vmhba> --key=<key> --value=<value>`
Discovery-level CHAP	`esxcli iscsi adapter discovery sendtarget param set --adapter=<vmhba> --key=<key> --value=<value> --address=<sendtarget_address>`
Target-level CHAP	`esxcli iscsi adapter target portal param set --adapter=<vmhba> --key=<key> --value=<value> --address=<address> --name=<iqn.name>`

7 After setup is complete, run `esxcli storage core adapter rescan --adapter=<iscsi_adapter>` to rescan all storage devices.

8 After setup is complete, perform rediscovery and rescan all storage devices.

The following example performs the rediscovery and rescan operations.

```
esxcli <conn_options> iscsi adapter discovery rediscover
esxcli <conn_options> storage core adapter rescan --adapter=vmhba36
```

Listing and Setting iSCSI Options

You can list and set iSCSI options with ESXCLI.

You can also manage parameters. See Listing and Setting iSCSI Parameters.

Listing iSCSI Options with ESXCLI

You can use `esxcli iscsi` information retrieval commands to list external HBA properties, information about targets, and LUNs.

You can use the following `esxcli iscsi` options to list iSCSI parameters. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- Run `esxcli iscsi adapter firmware` to list or upload the firmware for the iSCSI adapter.

  ```
  esxcli <conn_options> iscsi adapter firmware get --adapter=<adapter_name>
  esxcli <conn_options> iscsi adapter firmware set --file=<firmware_file_path>
  ```

  The system returns information about the vendor, model, description, and serial number of the HBA.

- Run commands in the `esxcli iscsi adapter target` name space.
  - `esxcli iscsi adapter target portal` lists and sets authentication and portal parameters.
  - `esxcli iscsi adapter target list` lists LUN information.

Setting MTU with ESXCLI

You can change MTU settings by using ESXCLI.

If you want to change the MTU used for your iSCSI storage, you must make the change in two places.

- Run `esxcli network vswitch standard set` to change the MTU of the virtual switch.
- Run `esxcli network ip interface set` to change the MTU of the network interface.

Listing and Setting iSCSI Parameters

You can list and set iSCSI parameters for software iSCSI and for dependent hardware iSCSI by using ESXCLI. You can also return parameters to default inheritance.

Listing and Setting iSCSI Parameters with ESXCLI

You can list and set iSCSI parameters for software iSCSI and for dependent hardware iSCSI by using ESXCLI.

You can retrieve and set iSCSI parameters by running one of the following commands.

Parameter Type	Command
Adapter-level parameters	`esxcli iscsi adapter param set --adapter=<vmhba> --key=<key> --value=<value>`
Target-level parameters	`esxcli iscsi adapter target portal param set --adapter=<vmhba> --key=<key> --value=<value> --address=<address> --name=<iqn.name>`
Discovery-level parameters	`esxcli iscsi adapter discovery sendtarget param set --adapter=<vmhba> --key=<key> --value=<value> --address=<address>`

The following table lists all settable parameters. These parameters are also described in the IETF rfc 3720. You can run `esxcli iscsi adapter param get` to determine whether a parameter is settable or not.

The parameters in the table apply to software iSCSI and dependent hardware iSCSI.

Table 5-4. Settable iSCSI Parameters

Parameter	Description
`DataDigestType`	Increases data integrity. When data digest is enabled, the system performs a checksum over each PDUs data part and verifies using the CRC32C algorithm. **Note** Systems that use Intel Nehalem processors offload the iSCSI digest calculations for software iSCSI, thus reducing the impact on performance. Valid values are `digestProhibited`, `digestDiscouraged`, `digestPreferred`, or `digestRequired`.
`HeaderDigest`	Increases data integrity. When header digest is enabled, the system performs a checksum over the header part of each iSCSI Protocol Data Unit (PDU) and verifies using the CRC32C algorithm.
`MaxOutstandingR2T`	Max Outstanding R2T defines the Ready to Transfer (R2T) PDUs that can be in transition before an acknowledgement PDU is received.
`FirstBurstLength`	Maximum amount of unsolicited data an iSCSI initiator can send to the target during the execution of a single SCSI command, in bytes.
`MaxBurstLength`	Maximum SCSI data payload in a Data-In or a solicited Data-Out iSCSI sequence, in bytes.
`MaxRecvDataSegLen`	Maximum data segment length, in bytes, that can be received in an iSCSI PDU.
`NoopOutInterval`	Time interval, in seconds, between NOP-Out requests sent from your iSCSI initiator to an iSCSI target. The NOP-Out requests serve as the ping mechanism to verify that a connection between the iSCSI initiator and the iSCSI target is active. Supported only at the initiator level.
`NoopOutTimeout`	Amount of time, in seconds, that can lapse before your host receives a NOP-In message. The message is sent by the iSCSI target in response to the NOP-Out request. When the `NoopTimeout` limit is exceeded, the initiator terminates the current session and starts a new one. Supported only at the initiator level.

Table 5-4. Settable iSCSI Parameters (continued)

Parameter	Description
`RecoveryTimeout`	Amount of time, in seconds, that can lapse while a session recovery is performed. If the timeout exceeds its limit, the iSCSI initiator terminates the session.
`DelayedAck`	Allows systems to delay acknowledgment of received data packets.

You can use the following ESXCLI commands to list parameter options.

- Run `esxcli iscsi adapter param get` to list parameter options for the iSCSI adapter.
- Run `esxcli iscsi adapter discovery sendtarget param get` or `esxcli iscsi adapter target portal param set` to retrieve information about iSCSI parameters and whether they are settable.
- Run `esxcli iscsi adapter discovery sendtarget param get` or `esxcli iscsi adapter target portal param set` to set iSCSI parameter options.

If special characters are in the `<name>=<value>` sequence, for example, if you add a space, you must surround the sequence with double quotes ("`<name> = <value>`").

Returning Parameters to Default Inheritance with ESXCLI

The values of iSCSI parameters associated with a dynamic discovery address or a static discovery target are inherited from the corresponding settings of the parent.

For the dynamic discovery address, the parent is the adapter. For the static target, the parent is the adapter or discovery address.

- If you use the vSphere Client to modify authentication settings, you must deselect the **Inherit from Parent** check box before you can make a change to the discovery address or discovery target.
- If you use `esxcli iscsi`, the value you set overrides the inherited value.

Inheritance is relevant only if you want to return a dynamic discovery address or a static discovery target to its inherited value. In that case, use the following command, which requires the `--name` option for static discovery addresses, but not for dynamic discovery targets.

Target Type	Command
Dynamic target	`esxcli iscsi adapter discovery sendtarget param set`
Static target	`esxcli iscsi adapter target portal param set`

Enabling iSCSI Authentication

You can enable iSCSI authentication by using ESXCLI.

Enable iSCSI Authentication with ESXCLI

You can use the `esxcli iscsi adapter auth` commands to enable iSCSI authentication.

For information on iSCSI CHAP, see Setting iSCSI CHAP.

Procedure

1 (Optional) Set the authentication information for CHAP.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name> --
chap_password=<pwd> --level=[prohibited, discouraged, preferred, required] --secret=<string> --
adapter=<adapter_name>
```

You can set per-target CHAP for static targets, per-adapter CHAP, or apply the command to the discovery address.

Option	Command
Per-adapter CHAP	`esxcli iscsi adapter auth chap set`
Per-discovery CHAP	`esxcli iscsi adapter discovery sendtarget auth chap set`
Per-target CHAP	`esxcli iscsi adapter target portal auth chap set`

The following example sets adapter-level CHAP.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=User1 --
chap_password=MySpecialPwd --level=preferred --secret=uni_secret --adapter=vmhba33
```

2 (Optional) Set the authentication information for mutual CHAP by running `esxcli iscsi adapter auth chap` set again with the `-d` option set to `mutual option` and a different authentication user name and secret.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=mutual --mchap_username=<m_name> --
mchap_password=<m_pwd> --level=[prohibited, required] --secret=<string> --adapter=<adapter_name>
```

For `<level>`, specify `prohibited` or `required`.

Option	Description
`prohibited`	The host does not use CHAP authentication. If authentication is enabled, specify `chapProhibited` to disable it.
`required`	The host requires successful CHAP authentication. The connection fails if CHAP negotiation fails. You can set this value for mutual CHAP only if CHAP is set to `chapRequired`.

For direction, specify `mutual`.

Important You are responsible for making sure that CHAP is set before you set mutual CHAP, and for using compatible levels for CHAP and mutual CHAP. Use a different secret in CHAP and mutual CHAP.

Enable Mutual iSCSI Authentication with ESXCLI

Mutual authentication is supported for software iSCSI and dependent hardware iSCSI, but not for independent hardware iSCSI.

For information on iSCSI CHAP, see Setting iSCSI CHAP.

Prerequisites

- Verify that CHAP authentication is already set up when you start setting up mutual CHAP.
- Verify that CHAP and mutual CHAP use different user names and passwords. The second user name and password are supported for mutual authentication on the storage side.
- Verify that CHAP and mutual CHAP use compatible CHAP levels.

Procedure

1 Enable authentication.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name> --
chap_password=<pw> --level=[prohibited, discouraged, preferred, required] --secret=<string> --
adapter=<adapter_name>
```

The specified `chap_username` and `secret` must be supported on the storage side.

2 List possible VMkernel NICs to bind.

```
esxcli <conn_options> iscsi logicalnetworkportal list
```

3 Enable mutual authentication.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=mutual --mchap_username=<m_name> --
mchap_password=<m_pwd> --level=[prohibited, required] --secret=<string> --adapter=<adapter_name>
```

The specified `mchap_username` and `secret` must be supported on the storage side.

4 After setup is complete, perform rediscovery and rescan all storage devices.

The following example performs the rediscovery and rescan operations.

```
esxcli <conn_options> iscsi adapter discovery rediscover
esxcli <conn_options> storage core adapter rescan --adapter=vmhba36
```

Set Up Ports for iSCSI Multipathing

With port binding, you create a separate VMkernel port for each physical NIC using 1:1 mapping.

You can add all network adapter and VMkernel port pairs to a single vSwitch. The *vSphere Storage* documentation explains in detail how to specify port binding.

In the examples below, specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Prerequisites

Verify that you are familiar with iSCSI session removal. See Removing iSCSI Sessions.

Procedure

1 Find out which uplinks are available for use with iSCSI adapters.

```
esxcli <conn_options> iscsi physicalnetworkportal list --adapter=<adapter_name>
```

2 Connect the software iSCSI or dependent hardware iSCSI initiator to the iSCSI VMkernel ports by running the following command for each port.

```
esxcli <conn_options> iscsi networkportal nic add --adapter=<adapter_name> --nic=<bound_nic>
```

3 Verify that the ports were added to the iSCSI initiator by running the following command.

```
esxcli <conn_options> iscsi networkportal list --adapter=<adapter_name>
```

4 (Optional) If there are active iSCSI sessions between your host and targets, discontinue them. See Removing iSCSI Sessions.

5 Rescan the iSCSI initiator.

```
esxcli <conn_options> storage core adapter rescan --adapter <iscsi adapter>
```

6 To disconnect the iSCSI initiator from the ports, run the following command.

```
esxcli <conn_options> iscsi networkportal remove --adapter=<adapter_name> --nic=<bound_nic>
```

Managing iSCSI Sessions

To communicate with each other, iSCSI initiators and targets establish iSCSI sessions. You can use `esxcli iscsi session` to list and manage iSCSI sessions for software iSCSI and dependent hardware iSCSI.

Introduction to iSCSI Session Management

By default, software iSCSI and dependent hardware iSCSI initiators start one iSCSI session between each initiator port and each target port.

If your iSCSI initiator or target has more than one port, your host can establish multiple sessions. The default number of sessions for each target equals the number of ports on the iSCSI adapter times the number of target ports. You can display all current sessions to analyze and debug them. You might add sessions to the default for several reasons.

- Cloning sessions - Some iSCSI arrays support multiple sessions between the iSCSI adapter and target ports. If you clone an existing session on one of these arrays, the array presents more data

paths for your adapter. Duplicate sessions do not persist across reboot. Additional sessions to the target might have performance benefits, but the result of cloning depends entirely on the array. You must log out from an iSCSI session if you want to clone a session. You can use the `esxcli iscsi session add` command to clone a session.

- Enabling Header and Data Digest - If you are logged in to a session and want to enable the `Header` and `Data Digest` parameters, you must set the parameter, remove the session, and add the session back for the parameter change to take effect. You must log out from an iSCSI session if you want to clone a session.
- Establishing target-specific sessions - You can establish a session to a specific target port. This can be useful if your host connects to a single-port storage system that, by default, presents only one target port to your initiator, but can redirect additional sessions to a different target port. Establishing a new session between your iSCSI initiator and another target port creates an additional path to the storage system.

Caution Some storage systems do not support multiple sessions from the same initiator name or endpoint. Attempts to create multiple sessions to such targets can result in unpredictable behavior of your iSCSI environment.

Listing iSCSI Sessions

You can use `esxcli iscsi session` to list sessions.

The following example scenario uses the available commands. Run `esxcli iscsi session --help` and each command with `--help` for reference information. The example uses a configuration file to log in to the host. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- List a software iSCSI session at the adapter level.

```
esxcli <conn_options> iscsi session list --adapter=<iscsi_adapter>
```

- List a software iSCSI session at the target level.

```
esxcli <conn_options> iscsi session list --name=<target> --adapter=<iscsi_adapter>
```

Logging in to iSCSI Sessions

You can use `esxcli iscsi session` to log in to a session.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- Log in to a session on the current software iSCSI or dependent hardware iSCSI configuration at the adapter level.

```
esxcli <conn_options> iscsi session add --adapter=<adapter_name>
```

The following example applies custom values.

```
esxcli --config /host-config-file iscsi session add --adapter=vmhba36
```

- Log in to a session on the current software iSCSI or dependent hardware iSCSI configuration at the target level.

  ```
  esxcli <conn_options> iscsi session add --name=<target> --adapter=<adapter_name>
  ```

  The following example applies custom values.

  ```
  esxcli --config /host-config-file iscsi session add -name=iqn.xxx --adapter=vmhba36
  ```

- Add duplicate sessions with target and session IDs in current software iSCSI or dependent hardware iSCSI configuration.

  ```
  esxcli <conn_options> iscsi session add --name=<iqn.xxxx> --isid=<session_id> --
  adapter=<iscsi_adapter>
  ```

  `iqn.xxxx` is the target IQN, which you can determine by listing all sessions. `session_id` is the session's iSCSI ID. The following example applies custom values.

  ```
  esxcli --config /host-config-file iscsi session add -name=iqn.xxx --isid='00:02:3d:00:00:01' --
  adapter=vmhba36
  ```

Removing iSCSI Sessions

You can use `esxcli iscsi session` to remove iSCSI sessions.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration at the adapter level.

  ```
  esxcli <conn_options> iscsi session remove --adapter=<iscsi_adapter>
  ```

  The following example applies custom values.

  ```
  esxcli iscsi session remove --adapter=vmhba36
  ```

- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration at the target level.

  ```
  esxcli <conn_options> iscsi session remove --name=<iqn> --adapter=<iscsi_adapter>
  ```

  The following example applies custom values.

  ```
  esxcli <conn_options> iscsi session remove --name=iqn.xxx --adapter=vmhba38
  ```

- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration with target and session ID.

  ```
  esxcli <conn_options> iscsi session remove --name=<iqn.xxxx> --isid=<session id> --
  adapter=<iscsi_adapter>
  ```

  `iqn.xxxx` is the target IQN, which you can determine by listing all sessions. `session_id` is the session's iSCSI ID. The following example applies custom values.

  ```
  esxcli --config /host-config-file iscsi session remove --name=iqn.xxx --
  session='00:02:3d:01:00:01' --adapter=vmhba36
  ```

Managing Third-Party Storage Arrays

6

VMware partners and customers can optimize performance of their storage arrays in conjunction with VMware vSphere by using VMware PSA (pluggable storage architecture). The `esxcli storage core` namespace manages VMware PSA and the `esxcli storage nmp` namespace manages the VMware NMP plug-in.

The *vSphere Storage* documentation discusses PSA functionality in detail and explains how to use the vSphere Client to manage the PSA, the associated native multipathing plug-in (NMP), and third-party plug-ins.

This chapter uses the following acronyms.

Acronym	Meaning
PSA	Pluggable Storage Architecture
NMP	Native Multipathing Plug-in. Generic VMware multipathing module.
PSP	Path Selection Plug-in. Handles path selection for a given device.
SATP	Storage Array Type Plug-in. Handles path failover for a given storage array.

This chapter includes the following topics:

Managing NMP with esxcli storage nmp

The NMP is an extensible multipathing module that ESXi supports by default. You can use `esxcli storage nmp` to manage devices associated with NMP and to set path policies.

The NMP supports all storage arrays listed on the VMware storage Hardware Compatibility List (HCL) and provides a path selection algorithm based on the array type. The NMP associates a set of physical paths with a storage device (LUN). An SATP determines how path failover is handled for a specific storage array. A PSP determines which physical path is used to issue an I/O request to a storage device. SATPs and PSPs are plug-ins within the NMP.

Device Management with esxcli storage nmp device

The device option performs operations on devices currently claimed by the VMware NMP.

esxcli storage nmp device list

The `list` command lists the devices controlled by VMware NMP and shows the SATP and PSP information associated with each device. To show the paths claimed by NMP, run `esxcli storage nmp path list` to list information for all devices, or for just one device with the `--device` option.

Options	Description
`--device <device>` `-d <device>`	Filters the output of the command to show information about a single device. Default is all devices.

esxcli storage nmp device set

The `set` command sets the PSP for a device to one of the policies loaded on the system.

Any device can use the PSP assigned to the SATP handling that device, or you can run `esxcli storage nmp device set --device naa.xxx --psp <psp>` to specifically override the PSP assigned to the device.

- If a device does not have a specific PSP set, it always uses the PSP assigned to the SATP. If the default PSP for the SATP changes, the PSP assigned to the device changes only after reboot or after a device is reclaimed. A device is reclaimed when you unclaim all paths for the device and reclaim the paths.
- If you use `esxcli storage nmp device set` to override the SATPs default PSP with a specific PSP, the PSP changes immediately and remains the user-defined PSP across reboots. A change in the SATP's PSP has no effect.

Use the `--default` option to return the device to using the SATP's PSP.

Options	Description
`--default` `-E`	Sets the PSP back to the default for the SATP assigned to this device.
`--device <device>` `-d <device>`	Device to set the PSP for.
`--psp <PSP>` `-P <PSP>`	PSP to assign to the specified device. Call `esxcli storage nmp psp list` to display all currently available PSPs. See Managing Path Policies. See *vSphere Storage* for a discussion of path policies.

To set the path policy for the specified device to `VMW_PSP_FIXED`, run the following command.

```
esxcli <conn_options> storage nmp device set --device naa.xxx --psp VMW_PSP_FIXED
```

Listing Paths with esxcli storage nmp path

You can use the `path` option to list paths claimed by NMP.

By default, the command displays information about all paths on all devices. You can filter in the following ways.

- Only show paths to a singe device.

```
esxcli storage nmp path list --device <device>
```

- Only show information for a single path.

```
esxcli storage nmp path list --path=<path>
```

To list devices, call `esxcli storage nmp device list`.

Managing Path Selection Policy Plug-Ins with esxcli storage nmp psp

You can use `esxcli storage nmp psp` to manage VMware path selection policy plug-ins included with the VMware NMP and to manage third-party PSPs.

Important When used with third-party PSPs, the syntax depends on the third-party PSP implementation.

Retrieving PSP Information

The `esxcli storage nmp psp generic deviceconfig get` and `esxcli storage nmp psp generic pathconfig get` commands retrieve PSP configuration parameters. The type of PSP determines which command to use.

- Use `nmp psp generic deviceconfig get` for PSPs that are set to `VMW_PSP_RR`, `VMW_PSP_FIXED` or `VMW_PSP_MRU`.
- Use `nmp psp generic pathconfig get` for PSPs that are set to `VMW_PSP_FIXED` or `VMW_PSP_MRU`. No path configuration information is available for `VMW_PSP_RR`.

To retrieve PSP configuration parameters, use the appropriate command for the PSP.

- Device configuration information.

```
esxcli <conn_options> storage nmp psp generic deviceconfig get --device=<device>
esxcli <conn_options> storage nmp psp fixed deviceconfig get --device=<device>
esxcli <conn_options> storage nmp psp roundrobin deviceconfig get --device=<device>
```

- Path configuration information.

```
esxcli <conn_options> storage nmp psp generic pathconfig get --path=<path>
```

- Retrieve the PSP configuration for the specified path.

```
esxcli <conn_options> nmp psp pathconfig generic get --path vmhba4:C1:T2:L23
```

The `esxcli storage nmp psp list` command shows the list of PSPs on the system and a brief description of each plug-in.

Setting Configuration Parameters for Third-Party Extensions

The `esxcli storage nmp psp generic deviceconfig set` and `esxcli storage nmp psp generic pathconfig set` commands support future third-party PSA expansion. The `setconfig` command sets PSP configuration parameters for those third-party extensions.

Note The precise results of these commands depend on the third-party extension. See the extension documentation for information.

Use `esxcli storage nmp roundrobin setconfig` for other path policy configuration. See Customizing Round Robin Setup.

You can run `esxcli storage nmp psp generic deviceconfig set --device=<device>` to specify PSP information for a device, and `esxcli storage nmp psp generic pathconfig set --path=<path>` to specify PSP information for a path. For each command, use `--config` to set the specified configuration string.

Options	Description
`--config <config_string>` `-c <config_string>`	Configuration string to set for the device or path specified by `--device` or `--path`. See Managing Path Policies.
`--device <device>` `-d <device>`	Device for which you want to customize the path policy.
`--path <path>` `-p <path>`	Path for which you want to customize the path policy.

Fixed Path Selection Policy Operations

The `fixed` option gets and sets the preferred path policy for NMP devices configured to use `VMW_PSP_FIXED`.

Retrieving the Preferred Path

The `esxcli storage nmp fixed deviceconfig get` command retrieves the preferred path on a specified device that is using NMP and the `VMW_PSP_FIXED` path policy.

Options	Description
--device <device> -d <device>	Device for which you want to get the preferred path. This device must be controlled by the `VMW_PSP_FIXED` PSP.

To return the path configured as the preferred path for the specified device, run the following command. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

```
esxcli <conn_options> storage nmp fixed deviceconfig get --device naa.xxx
```

Setting the Preferred Path

The `esxcli storage nmp fixed deviceconfig set` command sets the preferred path on a specified device that is using NMP and the `VMW_PSP_FIXED` path policy.

Options	Description
`--device <device>` `-d <device>`	Device for which you want to set the preferred path. This device must be controlled by the `VMW_PSP_FIXED` PSP. Use `esxcli storage nmp device --list` to list the policies for all devices.
`--path <path>` `-p <path>`	Path to set as the preferred path for the specified device.

To set the preferred path for the specified device to `vmhba3:C0:T5:L3`, run the following command. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

```
esxcli <conn_options> storage nmp fixed deviceconfig set --device naa.xxx --path vmhba3:C0:T5:L3
```

Customizing Round Robin Setup

You can use the `esxcli storage nmp psp roundrobin` commands to set round robin path options on a device controlled by the `VMW_PSP_RR` PSP.

Specifying and Customizing Round Robin Path Policies

You can use `esxcli storage nmp` commands to set path policies. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

1 Set the path policy to round robin.

```
esxcli <conn_options> storage nmp device set --device naa.xxx --psp VMW_PSP_RR
```

2 Specify when to switch paths.

You can choose the number of I/O operations, number of bytes, and so on. The following example sets the device specified by `--device` to switch to the next path each time 12345 bytes have been sent along the current path.

```
esxcli <conn_options> storage nmp psp roundrobin deviceconfig set --type "bytes" -B 12345 --
device naa.xxx
```

The following example sets the device specified by `--device` to switch after 4200 I/O operations have been performed on a path.

```
esxcli <conn_options> storage nmp psp roundrobin deviceconfig set --type=iops --iops 4200 --
device naa.xxx
```

Retrieving Path Selection Settings

The `esxcli storage nmp psp roundrobin deviceconfig get` command retrieves path selection settings for a device that is using the `roundrobin` PSP. You can specify the device to retrieve the information for.

Options	Description
`-d <device>` `--device <device>`	Device to get round robin properties for.

Specifying Conditions for Path Changes

The `esxcli storage nmp psp roundrobin deviceconfig set` command specifies under which conditions a device that is using the `VMW_PSP_RR` PSP changes to a different path. You can use `--bytes` or `--iops` to specify when the path should change.

Options	Description
`--bytes` `-B`	Number of bytes to send along one path for this device before the PSP switches to the next path. You can use this option only when `--type` is set to bytes.
`--device` `-d`	Device to set round robin properties for. This device must be controlled by the round robin (`VMW_PSP_RR`) PSP.
`--iops` `-I`	Number of I/O operations to send along one path for this device before the PSP switches to the next path. You can use this option only when `--type` is set to `iops`.
`--type` `-t`	Type of round robin path switching to enable for this device. The following values for `type` are supported. ■ `bytes`: Sets the trigger for path switching based on the number of bytes sent down a path. ■ `default`: Sets the trigger for path switching back to default values. ■ `iops`: Sets the trigger for path switching based on the number of I/O operations on a path. An equal sign (=) before the type or double quotes around the type are optional.
`--useANO` `-U`	If set to 1, the round robin PSP includes paths in the active, unoptimized state in the round robin set. If set to 0, the PSP uses active, unoptimized paths only if no active optimized paths are available. Otherwise, the PSP includes only active optimized paths in the round robin path set.

Managing SATPs

The `esxcli storage nmp satp` commands manage SATPs.

You can use these commands to perform the following tasks.

- Retrieve and set configuration parameters.
- Add and remove rules from the list of claim rules for a specified SATP.
- Set the default PSP for a specified SATP.
- List SATPs that are currently loaded into NMP and the associated claim rules.

The default SATP for an active-active FC array with a vendor and model not listed in the SATP rules is `VMW_SATP_DEFAULT_AA`.

Retrieving Information About SATPs

The `esxcli storage nmp satp list` command lists the SATPs that are currently available to the NMP system and displays information about those SATPs. This command supports no options and displays information about these SATPs.

```
esxcli <conn_options> storage nmp satp list
```

The rule list command lists the claim rules for SATPs.

```
esxcli <conn_options> storage nmp satp rule list
```

Adding SATP Rules

Claim rules specify that a storage device that uses a certain driver or transport or has a certain vendor or model should use a certain SATP. The `esxcli storage nmp satp rule add` command adds a rule that performs such a mapping to the list of claim rules. The options you specify define the rule. For example, the following command specifies that if a path has vendor `VMWARE` and model `Virtual`, the PSA assigns it to the `VMW_SATP_LOCAL` SATP.

```
esxcli <conn_options> storage nmp satp rule add --satp="VMW_SATP_LOCAL" --vendor="VMWARE" --
model="Virtual" --description="VMware virtual disk"
```

Option	Description
`--driver` `-D`	Driver string to set when adding the SATP claim rule.
`--device` `-d`	Device to set when adding SATP claim rules. Device rules are mutually exclusive with vendor/model and driver rules.
`--force` `-f`	Force claim rules to ignore validity checks and install the rule even if checks fail.
`--model` `-M`	Model string to set when adding the SATP claim rule. Can be the model name or a pattern `^mod*`, which matches all devices that start with `mod`. That is, the pattern successfully matches `mod1` and `modz`, but not `mymod1`. The command supports the start/end (^) and wildcard (*) functionality but no other regular expressions.
`--transport` `-R`	Transport string to set when adding the SATP claim rule. Describes the type of storage HBA, for example, `iscsi` or `fc`.
`--vendor` `-V`	Vendor string to set when adding the SATP claim rule.
`--satp` `-s`	SATP for which the rule is added.
`--claim-option` `-c`	Claim option string to set when adding the SATP claim rule.
`--description` `-e`	Description string to set when adding the SATP claim rule.

Option	Description
--option -o	Option string to set when adding the SATP claim rule. Surround the option string in double quotes, and use a space, not a comma, when specifying more than one option. `"enable_local enable_ssd"`
--psp -P	Default PSP for the SATP claim rule.
--psp-option -O	PSP options for the SATP claim rule.
--type -t	Set the claim type when adding a SATP claim rule.

The following examples illustrate adding SATP rules. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- Add an SATP rule that specifies that disks with vendor string `VMWARE` and model string `Virtual` should be added to `VMW_SATP_LOCAL`.

  ```
  esxcli <conn_options> storage nmp satp rule add --satp="VMW_SATP_LOCAL" --vendor="VMWARE" --
  model="Virtual" --description="VMware virtual disk"
  ```

- Add an SATP rule that specifies that disks with the driver string `somedriver` should be added to `VMW_SATP_LOCAL`.

  ```
  esxcli <conn_options> storage nmp satp rule add --satp="VMW_SATP_LOCAL" --driver="somedriver"
  ```

- Add a rule that specifies that all storage devices with vendor string `ABC` and a model name that starts with `120` should use `VMW_SATP_DEFAULT_AA`.

  ```
  esxcli <conn_options> storage nmp satp rule add --satp VMW_SATP_DEFAULT_AA --vendor="ABC" --
  model="^120*
  ```

Removing SATP Rules

The `esxcli storage nmp satp rule remove` command removes an existing SATP rule. The options you specify define the rule to remove. The options listed for Adding SATP Rules are supported.

The following example removes the rule that assigns devices with vendor string `VMWARE` and model string `Virtual` to `VMW_SATP_LOCAL`.

```
esxcli <conn_options> storage nmp satp rule remove --satp="VMW_SATP_LOCAL" --vendor="VMWARE" --
model="Virtual"
```

Retrieving and Setting SATP Configuration Parameters

The `esxcli storage nmp satp generic deviceconfig get` and `esxcli storage nmp satp generic pathconfig get` commands retrieve per-device or per-path SATP configuration parameters. You cannot retrieve paths or devices for all SATPs, you must retrieve the information one path or one device at a time.

Use the following command to retrieve per device or per path SATP configuration parameters, and to see whether you can set specific configuration parameters for a device or path.

For example, `esxcli storage nmp satp generic deviceconfig get --device naa.xxx` might return SATP `VMW_SATP_LSI does not support device configuration`.

```
esxcli storage nmp satp generic pathconfig get -path vmhba1:C0:T0:L8 might return INIT,AVT
OFF,v5.4,DUAL ACTIVE,ESX FAILOVER
```

The `esxcli storage nmp satp generic deviceconfig set` and `esxcli storage nmp satp generic pathconfig set` commands set configuration parameters for SATPs that are loaded into the system, if they support device configuration. You can set per-path or per-device SATP configuration parameters.

Important The command passes the configuration string to the SATP associated with that device or path.

The configuration strings might vary by SATP. VMware supports a fixed set of configuration strings for a subset of its SATPs. The strings might change in future releases.

Options	Description
`--config` `-c`	Configuration string to set for the path specified by `--path` or the device specified by `--device`. You can set the configuration for the following SATPs. ■ `VMW_SATP_ALUA_CX` ■ `VMW_SATP_ALUA` ■ `VMW_SATP_CX` ■ `VMW_SATP_INV` You can specify one of the following device configuration strings. ■ `navireg_on` – starts automatic registration of the device with Navisphere. ■ `navireg_off` – stops the automatic registration of the device. ■ `ipfilter_on` – stops the sending of the host name for Navisphere registration. Used if host is known as `localhost`. ■ `ipfilter_off` – enables the sending of the host name during Navisphere registration.
`--device` `-d`	Device to set SATP configuration for. Not all SATPs support the setconfig option on devices.
`--path` `-p`	Path to set SATP configuration for. Not all SATPs support the setconfig option on paths.

Run `esxcli storage nmp device set --default --device=<device>` to set the PSP for the specified device back to the default for the assigned SATP for this device.

Path Claiming with esxcli storage core claiming

The `esxcli storage core claiming` namespace includes a number of troubleshooting commands.

These commands are not persistent and are useful only to developers who are writing PSA plug-ins or troubleshooting a system. If I/O is active on the path, unclaim and reclaim actions fail.

Important The help for `esxcli storage core claiming` includes the `autoclaim` command. Do not use this command unless instructed to do so by VMware support staff.

Using the Reclaim Troubleshooting Command

The `esxcli storage core claiming reclaim` troubleshooting command is intended for PSA plug-in developers or administrators who troubleshoot PSA plug-ins.

The command performs the following tasks.

- Attempts to unclaim all paths to a device.
- Runs the loaded claim rules on each of the unclaimed paths to reclaim those paths.

It is normal for this command to fail if a device is in use.

Important The reclaim command unclaims paths associated with a device.

You cannot use the command to reclaim paths currently associated with the `MASK_PATH` plug-in because `--device` is the only option for reclaim and `MASK_PATH` paths are not associated with a device.

You can use the command to unclaim paths for a device and have those paths reclaimed by the `MASK_PATH` plug-in.

Options	Description
`--device <device>` `-d <device>`	Name of the device on which all paths are reclaimed.
`--help`	Displays the help message.

Unclaiming Paths or Sets of Paths

The `esxcli storage core claiming unclaim` command unclaims a path or set of paths, disassociating those paths from a PSA plug-in. The command fails if the device is in use.

You can unclaim only active paths with no outstanding requests. You cannot unclaim the ESXi USB partition or devices with VMFS volumes on them. It is therefore normal for this command to fail, especially when you specify a plug-in or adapter to unclaim.

Unclaiming does not persist. Periodic path claiming reclaims unclaimed paths unless claim rules are configured to mask a path. See the *vSphere Storage* documentation for details.

Important The `unclaim` command unclaims paths associated with a device. You can use this command to unclaim paths associated with the `MASK_PATH` plugin but cannot use the `--device` option to unclaim those paths.

Options	Description
`--adapter <adapter>` `-A <adapter>`	If `--type` is set to `location`, specifies the name of the HBA for the paths that you want to unclaim. If you do not specify this option, unclaiming runs on paths from all adapters.
`--channel <channel>` `-C <channel>`	If `--type` is set to `location`, specifies the SCSI channel number for the paths that you want to unclaim. If you do not specify this option, unclaiming runs on paths from all channels.
`--claimrule-class <cl>` `-c <cl>`	Claim rule class to use in this operation. You can specify `MP` (Multipathing), `Filter`, or `VAAI`. `Multipathing` is the default. `Filter` is used only for `VAAI`. Specify claim rules for both `VAAI_FILTER` and `VAAI` plug-in to use it.
`--device <device>` `-d <device>`	If `--type` is set to `device`, attempts to unclaim all paths to the specified device. If there are active I/O operations on the specified device, at least one path cannot be unclaimed.
`--driver <driver>` `-D <driver>`	If `--type` is `driver`, unclaims all paths specified by this HBA driver.
`--lun <lun_number>` `-L <lun_number>`	If `--type` is `location`, specifies the SCSI LUN for the paths to unclaim. If you do not specify `--lun`, unclaiming runs on paths with any LUN number.
`--model <model>` `-m <model>`	If `--type` is `vendor`, attempts to unclaim all paths to devices with specific model information (for multipathing plug-ins) or unclaim the device itself (for filter plug-ins). If there are active I/O operations on this device, at least one path fails to unclaim.
`--path <path>` `-p <path>`	If `--type` is `path`, unclaims a path specified by its path UID or runtime name.
`--plugin <plugin>` `-P`	If `--type` is `plugin`, unclaims all paths for a specified multipath plug-in. `<plugin>` can be any valid PSA plug-in on the system. By default, only `NMP` and `MASK_PATH` are available, but additional plugi-ns might be installed.
`--target <target>` `-T <target>`	If `--type` is `location`, unclaims the paths with the SCSI target number specified by target. If you do not specify `--target`, unclaiming runs on paths from all targets.
`--type <type>` `-t <type>`	Type of unclaim operation to perform. Valid values are `location`, `path`, `driver`, `device`, `plugin`, and `vendor`.
`--vendor <vendor>` `-v <vendor>`	If `--type` is `vendor`, attempts to unclaim all paths to devices with specific vendor info for multipathing plug-ins or unclaim the device itself for filter plug-ins. If there are any active I/O operations on this device, at least one path fails to unclaim

The following troubleshooting command tries to unclaim all paths on `vmhba1`.

```
esxcli <conn_options> storage core claiming unclaim --type location -A vmhba1
```

If a path is the last path to a device that was in use, or a if a path was very recently in use, the unclaim operation might fail. An error is logged that not all paths could be unclaimed. You can stop processes that might use the device and wait 15 seconds to let the device be quiesced, then retry the command.

Managing Claim Rules

The PSA uses claim rules to determine which multipathing module should claim the paths to a particular device and to manage the device. `esxcli storage core claimrule` manages claim rules.

Change the Current Claim Rules in the VMkernel

Claim rule modification commands do not operate on the VMkernel directly. Instead, they operate on the configuration file by adding and removing rules.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1. Run one or more of the `esxcli storage core claimrule` modification commands.

   For example, `add`, `remove`, or `move`.

2. Run `esxcli storage core claimrule load` to replace the current rules in the VMkernel with the modified rules from the configuration file.

What to do next

You can also run `esxcli storage core plugin list` to list all loaded plug-ins.

Adding Claim Rules

The `esxcli storage core claimrule add` command adds a claim rule to the set of claim rules on the system.

You can use this command to add new claim rules or to mask a path using the `MASK_PATH` claim rule. You must load the rules after you add them.

Options	Description
`--adapter <adapter>` -A <adapter>	Adapter of the paths to use. Valid only if `--type` is `location`.
`--autoassign` `-u`	Adds a claim rule based on its characteristics. The rule number is not required.
`--channel <channel>` `-C <channel>`	Channel of the paths to use. Valid only if `--type` is `location`.
`--claimrule-class <cl>` `-c <cl>`	Claim rule class to use in this operation. You can specify `MP` (default), `Filter`, or `VAAI`. To configure hardware acceleration for a new array, add two claim rules, one for the VAAI filter and another for the VAAI plug-in. See *vSphere Storage* for detailed instructions.
`--driver <driver>` `-D <driver>`	Driver for the HBA of the paths to use. Valid only if `--type` is `driver`.
`--force` `-f`	Force claim rules to ignore validity checks and install the rule.
`--lun <lun_number>` `-L <lun_number>`	LUN of the paths to use. Valid only if `--type` is `location`.
`--model <model>` `-M <model>`	Model of the paths to use. Valid only if `--type` is `vendor`. Valid values are values of the Model string from the SCSI inquiry string.

Options	Description
`--plugin` `-P`	PSA plug-in to use. Currently, the values are `NMP` or `MASK_PATH`, but third parties can ship their own PSA plug-ins in the future. `MASK_PATH` refers to the plug-in `MASK_PATH_PLUGIN`. The command adds claim rules for this plug-in if the user wants to mask the path. You can add a claim rule that causes the `MASK_PATH_PLUGIN` to claim the path to mask a path or LUN from the host. See the *vSphere Storage* documentation for details.
`--rule <rule_ID>` `-r <rule_ID>`	Rule ID to use. Run `esxcli storage core claimrule list` to see the rule ID. The rule ID indicates the order in which the claim rule is to be evaluated. User-defined claim rules are evaluated in numeric order starting with 101.
`--target <target>` `-T <target>`	Target of the paths to use. Valid only if `--type` is `location`.
`--transport <transport>` `-R <transport>`	Transport of the paths to use. Valid only if `--type` is `transport`. The following values are supported. ■ `block` – block storage ■ `fc` – FibreChannel ■ `iscsivendor` — iSCSI ■ `iscsi` – not currently used ■ `ide` — IDE storage ■ `sas` — SAS storage ■ `sata` — SATA storage ■ `usb` – USB storage ■ `parallel` – parallel ■ `unknown`
`--type <type>` `-t <type>`	Type of matching to use for the operation. Valid values are `vendor`, `location`, `driver`, and `transport`.
`--vendor` `-V`	Vendor of the paths to use. Valid only if `--type` is `vendor`. Valid values are values of the vendor string from the SCSI inquiry string.
`--wwnn`	World-Wide Node Number for the target to use in this operation.
`--wwpn`	World-Wide Port Number for the target to use in this operation.
`--xcopy-max-transfer-size` `-m`	Maximum data transfer size when using XCOPY. Valid only if `--xcopy-use-array-values` is specified.
`--xcopy-use-array-values` `-a`	Use the array reported values to construct the XCOPY command to be sent to the storage array. This applies to VAAI claim rules only.
`--xcopy-use-multi-segs` `-s`	Use multiple segments when issuing an XCOPY request. Valid only if `--xcopy-use-array-values` is specified.

Claim rules are numbered as follows.

- Rules 0–100 are reserved for internal use by VMware.
- Rules 101–65435 are available for general use. Any third-party multipathing plug-ins installed on your system use claim rules in this range. By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not remove this rule, unless you want to unmask these devices.
- Rules 65436–65535 are reserved for internal use by VMware.

When claiming a path, the PSA runs through the rules starting from the lowest number and determines is a path matches the claim rule specification. If the PSA finds a match, it gives the path to the corresponding plug-in. This is worth noticing because a given path might match several claim rules.

The following examples illustrate adding claim rules. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- Add rule 321, which claims the path on adapter vmhba0, channel 0, target 0, LUN 0 for the NMP.

```
esxcli <conn_options> storage core claimrule add –r 321 –t location –A vmhba0 –C 0 –T 0 –L 0 –P
NMP
```

- Add rule 429, which claims all paths provided by an adapter with the mptscsi driver for the `MASK_PATH` plug-in.

```
esxcli <conn_options> storage core claimrule add –r 429 –t driver –D mptscsi –P MASK_PATH
```

- Add rule 914, which claims all paths with vendor string `VMWARE` and model string `Virtual` for the NMP.

```
esxcli <conn_options> storage core claimrule add –r 914 –t vendor –V VMWARE –M Virtual –P NMP
```

- Add rule 1015, which claims all paths provided by FC adapters for the NMP.

```
esxcli <conn_options> storage core claimrule add –r 1015 –t transport –R fc –P NMP
```

Removing Claim Rules

The `esxcli storage core claimrule remove` command removes a claim rule from the set of claim rules on the system.

Important By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not remove this rule, unless you want to unmask these devices.

Option	Description
`––rule <rule_ID>` `–r <rule_ID>`	ID of the rule to be removed. Run `esxcli storage core claimrule list` to see the rule ID.

The following example removes rule 1015. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

```
esxcli <conn_options> storage core claimrule remove –r 1015
```

Listing Claim Rules

The `list` command lists all claim rules on the system.

You can specify the claim rule class as an argument.

Option	Description
`--claimrule-class <cl>` `-c <cl>`	Claim rule class to use in this operation. You can specify `MP` (Multipathing), `Filter`, or `VAAI`. Multipathing is the default. Filter is used only for VAAI. Specify claim rules for both `VAAI_FILTER` and `VAAI` plug-in to use it. See *vSphere Storage* for information about VAAI.

You can run the command as follows. The equal sign is optional, so both forms of the command have the same result. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

```
esxcli <conn_options> storage core claimrule list -c Filter
esxcli <conn_options> storage core claimrule list --claimrule-class=Filter
```

Loading Claim Rules

The `esxcli storage core claimrule load` command loads claim rules from the `esx.conf` configuration file into the VMkernel. Developers and experienced storage administrators might use this command for boot time configuration.

`esxcli storage core claimrule load` has no options. The command always loads all claim rules from `esx.conf`.

Moving Claim Rules

The `esxcli storage core claimrule move` command moves a claim rule from one rule ID to another.

Options	Description
`--claimrule-class <cl>` `-c <cl>`	Claim rule class to use in this operation.
`--new-rule <rule_ID>` `-n <rule_ID>`	New rule ID you want to give to the rule specified by the `--rule` option.
`--rule <rule_ID>` `-r <rule_ID>`	ID of the rule to be removed. Run `esxcli storage core claimrule list` to display the rule ID.

The following example renames rule 1016 to rule 1015 and removes rule 1016. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

```
esxcli <conn_options> storage core claimrule move -r 1015 -n 1016
```

Load and Apply Path Claim Rules

You can run the `esxcli storage core claimrule run` command to apply claim rules that are loaded.

If you do not call run, the system checks for claim rule updates every five minutes and applies them. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 Modify rules and load them.

```
esxcli <conn_options> storage core claimrule load
```

2 Quiesce the devices that use paths for which you want to change the rule and unclaim those paths.

```
esxcli <conn_options> storage core claiming unclaim --device=<device>
```

3 Run path claiming rules.

```
esxcli <conn_options> storage core claimrule run
```

Running Path Claim Rules

The `esxcli storage core claimrule run` command runs path claiming rules.

You can run this command to apply claim rules that are loaded. See Load and Apply Path Claim Rules.

You can also use the `esxcli storage core claimrule run` command for troubleshooting and boot time configuration.

Options	Description
`--adapter <adapter>` `-A <adapter>`	If `--type` is `location`, name of the HBA for the paths to run the claim rules on. To run claim rules on paths from all adapters, omit this option.
`--channel <channel>` `-C <channel>`	If `--type` is `location`, value of the SCSI channel number for the paths to run the claim rules on. To run claim rules on paths with any channel number, omit this option.
`--claimrule-class` `-c`	Claim rule class to use in this operation.
`--device` `-d`	Device UID to use for this operation.
`--lun <lun>` `-L <lun>`	If `--type` is `location`, value of the SCSI LUN for the paths to run claim rules on. To run claim rules on paths with any LUN, omit this option.
`--path <path_UID>` `-p <path_UID>`	If `--type` is `path`, this option indicates the unique path identifier (UID) or the runtime name of a path to run claim rules on.
`--target <target>` `-T <target>`	If `--type` is `location`, value of the SCSI target number for the paths to run claim rules on. To run claim rules on paths with any target number, omit this option.

Options	Description
`--type <location|path|all>` `-t <location|path|all>`	Type of claim to perform. By default, uses `all`, which means claim rules run without restriction to specific paths or SCSI addresses. Valid values are `location`, `path`, and `all`.
`--wait` `-w`	You can use this option only if you also use `--type all`. If the option is included, the claim waits for paths to settle before running the claim operation. In that case, the system does not start the claiming process until it is likely that all paths on the system have appeared before starting the claim process. After the claiming process has started, the command does not return until device registration has completed. If you add or remove paths during the claiming or the discovery process, this option might not work correctly.

Managing Users

7

An ESXi system grants access to its resources when a known user with appropriate permissions logs on to the system with a password that matches the one stored for that user.

You can use the vSphere SDK for all user management tasks. You cannot create ESXi users by using the vSphere Client.

This chapter includes the following topics:

Users in the vSphere Environment

Users and roles control who has access to vSphere components and what actions each user can perform.

User management is discussed in detail in the *vSphere Security* documentation.

vCenter Server and ESXi systems authenticate a user with a combination of user name, password, and permissions. Servers and hosts maintain lists of authorized users and the permissions assigned to each user.

Privileges define basic individual rights that are required to perform actions and retrieve information. ESXi and vCenter Server use sets of privileges, or roles, to control which users can access particular vSphere objects. ESXi and vCenter Server provide a set of pre-established roles.

The privileges and roles assigned on an ESXi host are separate from the privileges and roles assigned on a vCenter Server system. When you manage a host by using a vCenter Server system, only the privileges and roles assigned through the vCenter Server system are available. You cannot create ESXi users by using the vSphere Client.

Assigning Permissions with ESXCLI

You can use ESXCLI commands to manage permissions.

A set of ESXCLI commands allows you to perform the following operations.

- Give permissions to local users and groups by assigning them one of the predefined roles.

- Give permissions to Active Directory users and groups if your ESXi host has been joined to an Active Directory domain by assigning them one of the predefined roles.

Important When you manage local users on your ESXi host, you are not affecting the vCenter Server users.

Example: Manage Permissions

You can list, remove, and set permissions for a user or group, as shown in the following example.

1 List permissions.

```
esxcli system permission list
```

The system displays permission information. The second column indicates whether the information is for a user or group.

```
Principal            Is Group  Role
-----------------------------------
ABCDEFGH\esx^admins  true      Admin
dcui                 false     Admin
root                 false     Admin
vpxuser              false     Admin
test1                false     ReadOnly
```

2 Set permissions for a user or group. Specify the ID of the user or group, and set the `--group` option to `true` to indicate a group. Specify one of three roles, `Admin`, `ReadOnly` or `NoAccess`.

```
esxcli system permission set --id test1 -r ReadOnly
```

3 Remove permissions for a user or group.

```
esxcli system permission unset --id test1
```

Account Management

You can manage accounts by using the following commands.

```
esxcli system account add
esxcli system account set
esxcli system account list
esxcli system account remove
```

Managing Virtual Machines

8

You can manage virtual machines by using the vSphere Client. You can use ESXCLI to list all running virtual machines and to stop a specific virtual machine.

This chapter includes the following topics:

Forcibly Stop a Virtual Machine with ESXCLI

You can use ESXCLI to stop a virtual machine forcibly.

In some cases, virtual machines do not respond to the normal shutdown or stop commands. In these cases, it might be necessary to forcibly shut down the virtual machines. Forcibly shutting down a virtual machine might result in guest operating system data loss and is similar to pulling the power cable on a physical machine.

You can forcibly stop virtual machines that are not responding to normal stop operation with the `esxcli vm process kill` command. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1. List all running virtual machines on the system to see the World ID of the virtual machine that you want to stop.

   ```
   esxcli <conn_options> vm process list
   ```

2. Stop the virtual machine by running the following command.

   ```
   esxcli <conn_options> vm process kill --type <kill_type> --world-id <ID>
   ```

The command supports three `--type` options. Try the types sequentially - `soft` before `hard`, `hard` before `force`. The following types are supported through the `--type` option.

Type	Description
soft	Gives the VMX process a chance to shut down cleanly, like `kill` or `kill -SIGTERM`.
hard	Stops the VMX process immediately, like `kill -9` or `kill -SIGKILL`.
force	Stops the VMX process when other options do not work.

What to do next

If all three options do not work, reboot your ESXi host to resolve the issue.

Managing vSphere Networking 9

The ESXCLI networking commands allow you to manage the vSphere network services.

You can connect virtual machines to the physical network and to each other and configure vSphere standard switches. Limited configuration of vSphere distributed switches is also supported. You can also set up your vSphere environment to work with external networks such as SNMP or NTP.

This chapter includes the following topics:

Introduction to vSphere Networking

At the core of vSphere Networking are virtual switches.

vSphere supports standard switches (VSS) and distributed switches (VDS). Each virtual switch has a preset number of ports and one or more port groups.

Virtual switches allow your virtual machines to connect to each other and to connect to the outside world.

- When two or more virtual machines are connected to the same virtual switch, and those virtual machines are also on the same port group or VLAN, network traffic between them is routed locally.
- When virtual machines are connected to a virtual switch that is connected to an uplink adapter, each virtual machine can access the external network through that uplink. The adapter can be an uplink connected to a standard switch or a distributed uplink port connected to a distributed switch.

Virtual switches allow your ESXi host to migrate virtual machines with VMware vMotion and to use IP storage through VMkernel network interfaces.

- Using vMotion, you can migrate running virtual machines with no downtime. You cannot enable vMotion with ESXCLI.
- IP storage refers to any form of storage that uses TCP/IP network communication as its foundation and includes iSCSI and NFS for ESXi. Because these storage types are network based, they can use the same VMkernel interface and port group.

The network services that the VMkernel provides (iSCSI, NFS, and vMotion) use a TCP/IP stack in the VMkernel. The VMkernel TCP/IP stack is also separate from the guest operating system's network stack. Each of these stacks accesses various networks by attaching to one or more port groups on one or more virtual switches.

Networking Using vSphere Standard Switches

vSphere standard switches allow you to connect virtual machines to the outside world.

Figure 9-1. Networking with vSphere Standard Switches

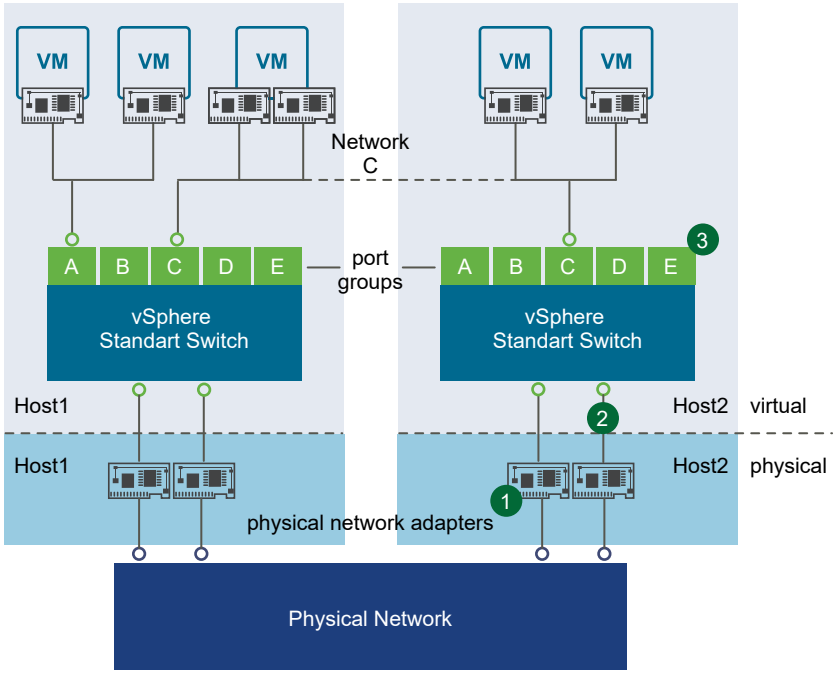


Figure 9-1. Networking with vSphere Standard Switches shows the relationship between the physical and virtual network elements. The numbers match those in the figure.

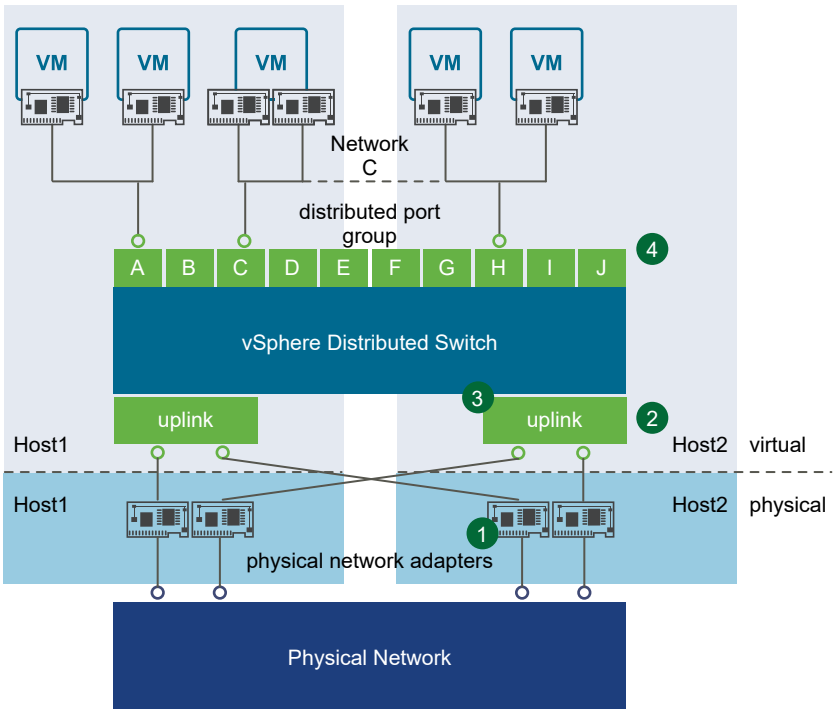
- Associated with each ESXi host are one or more uplink adapters (1). Uplink adapters represent the physical switches the ESXi host uses to connect to the network. You can manage uplink adapters by using the `esxcli network nic` command. See Managing Uplink Adapters.
- Each uplink adapter is connected to a standard switch (2). You can manage a standard switch and associate it with uplink adapters by using the `esxcli network vswitch` command. See Setting Up Virtual Switches and Associating a Switch with a Network Interface.

- Associated with the standard switch are port groups (3). Port group is a unique concept in the virtual environment. You can configure port groups to enforce policies that provide enhanced networking security, network segmentation, better performance, high availability, and traffic management. You can use the `esxcli network vswitch standard portgroup` command to associate a standard switch with a port group, and the `esxcli network ip interface` command to associate a port group with a VMkernel network interface.
- The VMkernel TCP/IP networking stack supports iSCSI, NFS, and vMotion and has an associated VMkernel network interface. You configure VMkernel network interfaces by using the `esxcli network ip interface` command. See Adding and Modifying VMkernel Network Interfaces. Separate VMkernel network interfaces are often used for separate tasks, for example, you might devote one VMkernel network interface card to vMotion only. Virtual machines run their own systems' TCP/IP stacks and connect to the VMkernel at the Ethernet level through virtual switches.

Networking Using vSphere Distributed Switches

When you want to connect a virtual machine to the outside world, you can use a standard switch or a distributed switch. With a distributed switch, the virtual machine can maintain its network settings even if the virtual machine is migrated to a different host.

Figure 9-2. Networking with vSphere Distributed Switches



- Each physical network adapter (1) on the host is paired with a distributed uplink port (2), which represents the uplink to the virtual machine. With distributed switches, the virtual machine no longer depends on the host's physical uplink but on the (virtual) uplink port. You manage a uplink ports primarily using the vSphere Client or vSphere APIs.

- The distributed switch itself (3) functions as a single virtual switch across all associated hosts. Because the switch is not associated with a single host, virtual machines can maintain consistent network configuration as they migrate from one host to another.

  Like a standard switch, each distributed switch is a network hub that virtual machines can use. A distributed switch can route traffic internally between virtual machines or link to an external network by connecting to physical network adapters. You create a distributed switch by using the vSphere Client UI. You can list distributed virtual switches by using the `esxcli network vswitch` command. See Setting Up Virtual Switches and Associating a Switch with a Network Interface.

Retrieving Basic Networking Information

Service console commands for retrieving networking information are not included in the ESXi Shell. You can instead use ESXCLI commands directly in the shell.

For information corresponding to the Linux `ifconfig` command, use the following ESXCLI commands.

```
esxcli <conn_options> network ip interface list
esxcli <conn_options> network ip interface ipv4 get –n vmk<X>
esxcli <conn_options> network ip interface ipv6 get –n vmk<X>
esxcli <conn_options> network ip interface ipv6 address list
```

For information corresponding to the Linux `netstat` command, use the following ESXCLI command.

```
esxcli <conn_options> network ip connection list
```

You can also ping individual hosts with the `esxcli network diag ping` command. The command includes options for using ICMPv4 or ICMPv6 packet requests, specifying an interface to use, specifying the interval, and so on.

Troubleshoot a Networking Setup

You can use ESXCLI network commands to view network statistics and troubleshoot your networking setup. The nested hierarchy of commands allows you to drill down to potential trouble spots.

Procedure

1 List all virtual machine networks on a host.

```
esxcli network vm list
```

The command returns for each virtual machine the World ID, name, number of ports, and networks, as in the following example.

```
World ID  Name                   Num Ports  Networks
--------------------------------------------------
   10374  ubuntu-server-11.04-1          2  VM Network, dvportgroup-19
   10375  ubuntu-server-11.04-2          2  VM Network, dvportgroup-19
   10376  ubuntu-server-11.04-3          2  VM Network, dvportgroup-19
   10408  ubuntu-server-11.04-4          3  VM Network, VM Network 10Gbps, dvportgroup-19
```

2 List the ports for one of the virtual machines by specifying its World ID.

```
esxcli network vm port list -w 10408
```

The command returns port information, as in the following example.

```
Port:
  Port ID: XXXXXXXX
  vSwitch: vSwitch0
  Portgroup: VM Network
  DVPort ID:
  MAC Address: 00:XX:XX:aa:XX:XX
  IP Address: 10.XXX.XXX.XXX
  Team Uplink: vmnic0
  Uplink Port ID: 12345678
  Active Filters:
```

3 Retrieve the switch statistics for a port.

```
esxcli network port stats get -p 12345678
```

The command returns detailed statistics, as in the following example.

```
Packet statistics for port 12345678:
   Packets received: 517631
   Packets sent: 18937
   Bytes received: 100471874
   Bytes sent: 1527233
   Broadcast packets received: 474160
   Broadcast packets sent: 107
   Multicast packets received: 8020
   Multicast packets sent: 8
   Unicast packets received: 35451
   Unicast packets sent: 18822
   Receive packets dropped: 45
   Transmit packets dropped: 0
```

4 Retrieve the filter information for the port.

```
esxcli network port filter stats get -p 12345678
```

The command returns detailed statistics, as in the following example.

```
Filter statistics for dvfilter-test:
   Filter direction: Receive
   Packets in: 202080
   Packets out: 202080
   Packets dropped: 0
   Packets filtered: 0
   Packets faulted: 0
   Packets queued: 0
   Packets injected: 0
   Packet errors: 0
```

5 Retrieve complete statistics for a NIC.

```
esxcli network nic stats get –n vmnic0
```

6 Get a per-VLAN packed breakdown on a NIC.

```
esxcli network nic vlan stats get –n vmnic0
```

The command returns the number of packets sent and received for the VLAN you specified.

Setting Up vSphere Networking with vSphere Standard Switches

You can use ESXCLI to set up the vSphere networking.

You can set up your virtual network by performing a set of tasks.

1 Create or manipulate virtual switches by using `esxcli network vswitch`. By default, each ESXi host has one virtual switch, `vSwitch0`. You can create additional virtual switches or manage existing switches. See Setting Up Virtual Switches and Associating a Switch with a Network Interface.

2 (Optional) Make changes to the uplink adapter by using `esxcli network vswitch standard uplink`. See Managing Uplink Adapters.

3 (Optional) Use `esxcli network vswitch standard portgroup` to add port groups to the virtual switch. See #unique_130.

4 (Optional) Use `esxcli network vswitch standard portgroup set` to establish VLANs by associating port groups with VLAN IDs. See #unique_131.

5 Use `esxcli network ip interface` to configure the VMkernel network interfaces. See Adding and Modifying VMkernel Network Interfaces.

Setting Up Virtual Switches and Associating a Switch with a Network Interface

A virtual switch models a physical Ethernet switch. You can manage virtual switches and port groups by using the vSphere Client or ESXCLI commands.

By default, each ESXi host has a single virtual switch called `vSwitch0`. Each virtual switch has logical ports. For information about maximum allowed virtual switches and ports, see the *VMware Configuration Maximums* tool. Ports connect to the virtual machines and the ESXi physical network adapters.

- You can connect one virtual machine network adapter to each port by using the vSphere Client UI.
- You can connect the uplink adapter to the virtual switches by using `esxcli network vswitch standard uplink`. See Linking and Unlinking Uplink Adapters with ESXCLI.

When two or more virtual machines are connected to the same virtual switch, network traffic between them is routed locally. If an uplink adapter is attached to the virtual switch, each virtual machine can access the external network that the adapter is connected to.

This section discusses working in a standard switch environment. See Networking Using vSphere Distributed Switches for information about distributed switch environments.

When working with virtual switches and port groups, perform the following tasks.

1. Find out which virtual switches are available and, optionally, what the associated MTU and CDP (Cisco Discovery Protocol) settings are. See Retrieving Information About Virtual Switches with ESXCLI.
2. Add a virtual switch. See Adding and Deleting Virtual Switches with ESXCLI.
3. For a newly added switch, perform these tasks.
   a. Add a port group. See Managing Port Groups with ESXCLI.
   b. (Optional) Set the port group VLAN ID. See Setting the Port Group VLAN ID with ESXCLI.
   c. Add an uplink adapter. See Linking and Unlinking Uplink Adapters with ESXCLI.
   d. (Optional) Change the MTU or CDP settings. See Setting Switch Attributes with ESXCLI.

Retrieving Information About Virtual Switches with ESXCLI

You can retrieve information about virtual switches by using `esxcli network vswitch` commands.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- List all virtual switches and associated port groups.

  ```
  esxcli <conn_options> network vswitch standard list
  ```

  The command prints information about the virtual switch, which might include its name, number of ports, MTU, port groups, and other information. The output includes information about CDP settings for the virtual switch. The precise information depends on the target system. The default port groups are `Management Network` and `VM Network`.

- List the network policy settings, such as security policy, traffic shaping policy, and failover policy, for the virtual switch. The following commands are supported.

```
esxcli <conn_options> network vswitch standard policy failover get
esxcli <conn_options> network vswitch standard policy security get
esxcli <conn_options> network vswitch standard policy shaping get
```

Adding and Deleting Virtual Switches

You can add and delete virtual switches with ESXCLI.

Adding and Deleting Virtual Switches with ESXCLI

You can add and delete virtual switches by using the `esxcli network vswitch standard` namespace.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- Add a virtual switch.

```
esxcli <conn_options> network vswitch standard add --vswitch-name=vSwitch42
```

  You can specify the number of port groups while adding the virtual switch. If you do not specify a value, the default value is used. The system-wide port count cannot be greater than 4096.

```
esxcli <conn_options> network vswitch standard add --vswitch-name=vSwitch42 --ports=8
```

  After you have added a virtual switch, you can set switch attributes. See Setting Switch Attributes with ESXCLI. You can also add one or more uplink adapters. See Linking and Unlinking Uplink Adapters with ESXCLI.

- Delete a virtual switch.

```
esxcli <conn_options> network vswitch standard remove --vswitch-name=vSwitch42
```

  You cannot delete a virtual switch if any ports on the switch are still in use by VMkernel networks or virtual machines. Run `esxcli network vswitch standard list` to determine whether a virtual switch is in use.

Setting Switch Attributes with ESXCLI

You can set the maximum transmission unit (MTU) and CDP status for a virtual switch. The CDP status shows which Cisco switch port is connected to which uplink.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- Set the MTU for a vSwitch.

```
esxcli <conn_options> network vswitch standard set --mtu=9000 --vswitch-name=vSwitch1
```

The MTU is the size, in bytes, of the largest protocol data unit the switch can process. When you set this option, it affects all uplinks assigned to the virtual switch.

- Set the CDP value for a vSwitch. You can set status to `down`, `listen`, `advertise`, or `both`.

```
esxcli <conn_options> network vswitch standard set --cdp-status=listen --vswitch-name=vSwitch1
```

Managing Port Groups with ESXCLI

You can use `esxcli network vswitch standard portgroup` to check, add, and remove port groups.

Network services connect to vSwitches through port groups. A port group allows you to group traffic and specify configuration options such as bandwidth limitations and VLAN tagging policies for each port in the port group. A virtual switch must have one port group assigned to it. You can assign additional port groups.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- List port groups currently associated with a virtual switch.

```
esxcli <conn_options> network vswitch standard portgroup list
```

The command lists the port group name, associated virtual switch, active clients, and VLAN ID.

- Add a port group.

```
esxcli <conn_options> network vswitch standard portgroup add --portgroup-name=<name> --vswitch-name=vSwitch1
```

- Delete one of the existing port groups.

```
esxcli <conn_options> network vswitch standard portgroup remove --portgroup-name=<name> --vswitch-name=vSwitch1
```

Connecting and Disconnecting Uplink Adapters and Port Groups with ESXCLI

You can use `esxcli network vswitch standard portgroup policy failover set` to connect and disconnect uplink adapters and port groups.

If your setup includes one or more port groups, you can associate each port group with one or more uplink adapters and remove the association. This functionality allows you to filter traffic from a port group to a specific uplink, even if the virtual switch is connected with multiple uplinks.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

- Connect a port group with an uplink adapter.

```
esxcli <conn_options> network vswitch standard portgroup policy failover set --active-uplinks=vmnic1,vmnic6,vmnic7
```

This command fails silently if the uplink adapter does not exist.

- Make some of the adapters standby instead of active.

```
esxcli <conn_options> network vswitch standard portgroup policy failover set --standby-
uplinks=vmnic1,vmnic6,vmnic7
```

Setting the Port Group VLAN ID with ESXCLI

You can use `esxcli network vswitch standard portgroup set` to manage VLANs.

VLANs allow you to further segment a single physical LAN segment so that groups of ports are isolated as if they were on physically different segments. The standard is IEEE 802.1Q.

A VLAN ID restricts port group traffic to a logical Ethernet segment within the physical network.

- Set the VLAN ID to 4095 to allow a port group to reach port groups located on other VLAN.
- Set the VLAN ID to 0 to disable the VLAN for this port group.

If you use VLAN IDs, you must change the port group labels and VLAN IDs together so that the labels properly represent connectivity. VLAN IDs are optional.

You can use the following commands for VLAN management.

- Allow port groups to reach port groups located on other VLANs.

```
esxcli <conn_options> network vswitch standard portgroup set -p <pg_name> --vlan-id 4095
```

Run the command multiple times to allow all ports to reach port groups located on other VLANs.

- Disable VLAN for port group g42.

```
esxcli <conn_options> network vswitch standard portgroup set --vlan-id 0 -p g42
```

Run `esxcli network vswitch standard portgroup list` to list all port groups and associated VLAN IDs.

Managing Uplink Adapters

You can manage uplink adapters, which represent the physical NICs that connect the ESXi host to the network by using the `esxcli network nic` command. You can also use `esxcli network vswitch` to link and unlink the uplink.

You can use `esxcli network nic` to list all uplinks, to list information, to set attributes, and to bring a specified uplink down or up.

Manage Uplink Adapters with ESXCLI

you can use to manage uplink adapters.

The following example workflow lists all uplink adapters, lists properties for one uplink adapter, changes the uplink's speed and duplex settings, and brings the uplink down and back up. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 List all uplinks and information about each device.

```
esxcli <conn_options> network nic list
```

You can narrow down the information displayed by using `esxcli network nic get --nic-name=<nic>`.

2 (Optional) Bring down one of the uplink adapters.

```
esxcli <conn_options> network nic down --nic-name=vmnic0
```

3 Change uplink adapter settings.

```
esxcli <conn_options> network nic set <option>
```

You must specify one of the following options.

Option	Description
-a|--auto	Sets the speed and duplex settings to autonegotiate.
-D|--duplex=<str>	Duplex to set this NIC to. Acceptable values are `full` and `half`.
-P | --phy-address	Sets the MAC address of the device
-l|--message-level=<long>	Sets the driver message level. Message levels and what they imply differ per driver.
-n|--nic-name=<str>	Name of the NIC to configured. Must be one of the cards listed in the `nic list` command (required).
-p|--port=<str>	Selects the device port. The following device ports are available. ■ `aui` ■ `bnc` ■ `fibre` ■ `mii` ■ `tp`
-S|--speed=<long>	Speed to set this NIC to. Acceptable values are 10, 100, 1000, and 10000.

Option	Description
-t|--transceiver-type=<str>	Selects transceiver type. The following transceiver types are available. ■ external ■ internal
-w|--wake-on-lan=<str>	Sets Wake-on-LAN options. Not all devices support this option. The option value is a string of characters specifying which options to enable. ■ p – Wake on phy activity ■ u – Wake on unicast messages ■ m – Wake on multicast messages ■ b – Wake on broadcast messages ■ a – Wake on ARP ■ g – Wake on MagicPacket ■ s – Enable SecureOn password for MagicPacket

4 (Optional) Bring the uplink adapter back up.

```
esxcli <conn_options> network nic up --nic-name=vmnic0
```

Specifying Multiple Uplinks with ESXCLI

At any time, one port group NIC array and a corresponding set of active uplinks exist. When you change the active uplinks, you also change the standby uplinks and the number of active uplinks.

The following example illustrates how active and standby uplinks are set.

1 The port group NIC array is `[vmnic1, vmnic0, vmnic3, vmnic5, vmnic6, vmnic7]` and `active-uplinks` is set to three uplinks - `vmnic1`, `vmnic0`, `vmnic3`. The other uplinks are standby uplinks.

2 You set the active uplinks to a new set `[vmnic3, vmnic5]`.

3 The new uplinks override the old set. The NIC array changes to `[vmnic3, vmnic5, vmnic6, vmnic7]`. `vmnic0` and `vmnic1` are removed from the NIC array and `max-active` becomes 2.

If you want to keep `vmnic0` and `vmnic1` in the array, you can make those NICs standby uplinks in the command that changes the active uplinks.

```
esxcli network vswitch standard portgroup policy failover set -p testPortgroup --active-uplinks
vmnic3,vmnic5 --standby-uplinks vmnic1,vmnic0,vmnic6,vmnic7
```

Linking and Unlinking Uplink Adapters with ESXCLI

You can use ESXCLI to link and unlink uplink adapters.

When you create a virtual switch by using `esxcli network vswitch standard add`, all traffic on that virtual switch is initially confined to that virtual switch. All virtual machines connected to the virtual switch can talk to each other, but the virtual machines cannot connect to the network or to virtual machines on other hosts. A virtual machine also cannot connect to virtual machines connected to a different virtual switch on the same host.

Having a virtual switch that is not connected to the network might make sense if you want a group of virtual machines to be able to communicate with each other, but not with other hosts or with virtual machines on other hosts. In most cases, you set up the virtual switch to transfer data to external networks by attaching one or more uplink adapters to the virtual switch.

You can use the following commands to list, add, and remove uplink adapters. When you link by using ESXCLI, the physical NIC is added as a standby adapter by default. You can then modify the teaming policy to make the physical NIC active by running the command `esxcli network vswitch standard policy failover set`.

- List uplink adapters.

  ```
  esxcli <conn_options> network vswitch standard list
  ```

  The uplink adapters are returned in the `Uplink` item.

- Add a new uplink adapter to a virtual switch.

  ```
  esxcli <conn_options> network vswitch standard uplink add --uplink-name=vmnic15 --vswitch-name=vSwitch0
  ```

- Remove an uplink adapter from a virtual switch.

  ```
  esxcli <conn_options> network vswitch standard uplink remove --uplink-name=vmnic15 --vswitch-name=vSwitch0
  ```

Adding and Modifying VMkernel Network Interfaces

VMkernel network interfaces are used primarily for management traffic, which can include vMotion, IP Storage, and other management traffic on the ESXi system. You can also bind a newly created VMkernel network interface for use by software and dependent hardware iSCSI by using the `esxcli iscsi` commands.

The VMkernel network interface is separate from the virtual machine network. The guest operating system and application programs communicate with a VMkernel network interface through a commonly available device driver or a VMware device driver optimized for the virtual environment. In either case, communication in the guest operating system occurs as it would with a physical device. Virtual machines can also communicate with a VMkernel network interface if both use the same virtual switch.

Each VMkernel network interface has its own MAC address and one or more IP addresses, and responds to the standard Ethernet protocol as would a physical NIC. The VMkernel network interface is created with TCP Segmentation Offload (TSO) enabled.

You can manage VMkernel NICs with ESXCLI.

Managing VMkernel Network Interfaces with ESXCLI

You can configure the VMkernel network interface for IPv4 or for IPv6 with ESXCLI.

For IPv4, see Add and Configure an IPv4 VMkernel Network Interface with ESXCLI. For IPv6, see Add and Configure an IPv6 VMkernel Network Interface with ESXCLI.

Add and Configure an IPv4 VMkernel Network Interface with ESXCLI

You can add and configure an IPv4 VMkernel NIC by using ESXCLI.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 Add a new VMkernel network interface.

```
esxcli <conn_options> network ip interface add --interface-name=vmk<X> --portgroup-
name=<my_portgroup>
```

You can specify the MTU setting after you have added the network interface by using `esxcli network ip interface set --mtu`.

2 Configure the interface as an IPv4 interface.

You must specify the IP address by using `--ip`, the netmask, and the name. For the following examples, assume that VMSF-VMK-363 is a port group to which you want to add a VMkernel network interface.

```
esxcli <conn_options> network ip interface ipv4 set --ipv4=<ip_address> --netmask=255.255.255.0 --
type=<value> --interface-name=vmk<X>
```

You can set the address as follows.

- `--ipv4=<X.X.X.X> --netmask=<X.X.X.X> --type=static` – Static IPv4 address.
- `--type=dhcp` – Use IPv4 DHCP.

The VMkernel supports DHCP only for ESXi 4.0 and later.

When the command finishes successfully, the newly added VMkernel network interface is enabled.

3 List information about all VMkernel network interfaces on the system.

```
esxcli <conn_options> network ip interface list
```

The command displays the network information, port group, MTU, and current state for each virtual network adapter in the system.

Add and Configure an IPv6 VMkernel Network Interface with ESXCLI

You can add and configure an IPv6 VMkernel NIC by using ESXCLI.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 Add a new VMkernel network interface.

```
esxcli <conn_options> network ip interface add --interface-name=vmk<x> --portgroup-
name=<my_portgroup>
```

You can specify the MTU setting after you have added the network interface by using `esxcli network ip interface set --mtu`.

When the command finishes successfully, the newly added VMkernel network interface is enabled.

2 Run `esxcli network ip interface ipv6 address add` to configure the interface as an IPv6 interface.

You must specify the IP address using `--ip` and the name. For the following examples, assume that VMSF-VMK-363 is a port group to which you want to add a VMkernel network interface.

```
esxcli <conn_options> network ip interface ipv6 address add --ip=<X:X:X::/X> --interface-
name=vmk<X>
```

You can set the address as follows.

- <X:X:X::/X> - Static IPv6 address.
- `--enable-dhcpv6` - Enables DHCPv6 on this interface and attempts to acquire an IPv6 address from the network.
- `--enable-router-adv` - Use the IPv6 address advertised by the router. The address is added when the router sends the next router advert.

The VMkernel supports DHCP only for ESXi 4.0 and later.

When the command finishes successfully, the newly added VMkernel network interface is enabled.

3 List information about all VMkernel network interfaces on the system.

```
esxcli <conn_options> network ip interface list
```

The command displays the network information, port group, MTU, and current state for each virtual network adapter in the system.

4 (Optional) Remove the IPv6 address and disable IPv6.

```
esxcli <conn_options> network ip interface ipv6 address remove --interface-name=<VMK_NIC> --
ipv6=<ipv6_addr>
esxcli <conn_options> network ip set --ipv6-enabled=false
```

Managing Standard Networking Services in the vSphere Environment

You can use ESXCLI commands to set up DNS, NTP, SNMP, and the default gateway for your vSphere environment.

Setting the DNS Configuration with ESXCLI

The `esxcli network ip dns` command lists and specifies the DNS configuration of your ESXi host.

Important If you try to change the host or domain name or the DNS server on hosts that use DHCP, an error results.

In network environments where a DHCP server and a DNS server are available, ESXi hosts are automatically assigned DNS names.

In network environments where automatic DNS is not available or you do not want to use automatic DNS, you can configure static DNS information, including a host name, primary name server, secondary name server, and DNS suffixes.

The `esxcli network ip dns` namespace includes two namespaces.

- `esxcli network ip dns search` includes commands for DNS search domain configuration.
- `esxcli network ip dns server` includes commands for DNS server configuration.

Set Up a DNS Server with ESXCLI

You can use ESXCLI to set up a DNS server.

The following example illustrates setting up a DNS server. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1. Print a list of DNS servers configured on the system in the order in which they will be used.

   ```
   esxcli <conn_options> network ip dns server list
   ```

   If DNS is not set up for the target server, the command returns an empty string.

2. Add a server by running `esxcli network ip dns server add` and specifying the server IPv4 or IPv6 address.

   ```
   esxcli <conn_options> network ip dns server add --server=<str>
   ```

3. Change the DNS settings.

   - Specify the DNS server by using the `--dns` option and the DNS host.

     ```
     esxcli <conn_options> network ip dns server add --server=<server>
     ```

     Run the command multiple times to specify multiple DNS hosts.

   - Configure the DNS host name for the server specified by `--server` or `--vihost`.

     ```
     esxcli <conn_options> system hostname set --host=<new_host_name>
     ```

- Configure the DNS domain name for the server specified by `--server` or `--vihost`.

```
esxcli <conn_options> system hostname --domain=mydomain.biz
```

4 To turn on DHCP, enable DHCP and set the VMkernel NIC.

- Turn on DHCP for IPv4.

```
esxcli <conn_options> network ip interface ipv4 set --type dhcp/none/static
esxcli <conn_options> network ip interface ipv4 set --peer-dns=<str>
```

- Turn on DHCP for IPv6.

```
esxcli <conn_options> network ip interface ipv6 set --enable-dhcpv6=true/false
esxcli <conn_options> network ip interface ipv6 set --peer-dns=<str>
```

Modify DNS Setup for a Preconfigured Server with ESXCLI

You can use ESXCLI to modify the setup of a preconfigured DNS server.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 Display DNS properties for the specified server.

a List the host and domain name.

```
esxcli <conn_options> system hostname get
```

b List available DNS servers.

```
esxcli <conn_options> network ip dns server list
```

c List the DHCP settings for individual VMkernel NICs.

```
esxcli <conn_options> network ip interface ipv4 get
```

```
esxcli <conn_options> network ip interface ipv6 get
```

2 If the DNS properties are set, and you want to change the DHCP settings, you must specify the virtual network adapter to use when overriding the system DNS.

You can override the existing DHCP setting by using the following commands.

```
esxcli <conn_options> network ip interface ipv4 set --type dhcp/none/static
esxcli <conn_options> network ip interface ipv6 set --enable-dhcpv6=true/false
```

Setting Up IPsec

You can set Internet Protocol Security by using `esxcli network ip ipsec`, which secures IP communications coming from and arriving at ESXi hosts. Administrators who perform IPsec setup must have a solid understanding of both IPv6 and IPsec.

ESXi hosts support IPsec only for IPv6 traffic, but not for IPv4 traffic.

You can run `esxcli network ip ipsec` commands with a vCenter Server system as a target, by using the `--vihost` option.

The VMware implementation of IPsec adheres to the following IPv6 RFCs.

- 4301 Security Architecture for the Internet Protocol
- 4303 IP Encapsulating Security Payload (ESP)
- 4835 Cryptographic Algorithm Implementation Requirements for ESP
- 2410 The NULL Encryption Algorithm and Its Use With IPsec
- 2451 The ESP CBC-Mode Cipher Algorithms
- 3602 The AES-CBC Cipher Algorithm and Its Use with IPsec
- 2404 The Use of HMAC-SHA-1-96 within ESP and AH
- 4868 Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512

Using IPsec with ESXi

When you set up IPsec on an ESXi host, you enable protection of incoming or outgoing data. What happens precisely depends on how you set up the system's Security Associations (SAs) and Security Policies (SPs).

- An SA determines how the system protects traffic. When you create an SA, you specify the source and destination, authentication, and encryption parameters, and an identifier for the SA with the following options.

esxcli network ip ipsec
`--sa-source` and `--sa-destination`
`--sa-spi`
`--sa-mode`
`--encryption-algorithm` and `--encryption-key`
`--integrity-algorithm` and `--integrity-key`

- An SP identifies and selects traffic that must be protected. An SP consists of two logical sections, a selector, and an action.

The selector is specified by the following options.

esxcli network ip ipsec
`--sa-source` and `--source-port`
`--destination-port`
`--upper-layer-protocol`
`--flow-direction`

The action is specified by the following options.

esxcli network ip ipsec
`--sa-name`
`--sp-name`
`--action`

Because IPsec allows you to target precisely which traffic should be encrypted, it is well suited for securing your vSphere environment. For example, you can set up the environment so all vMotion traffic is encrypted.

Managing Security Associations

You can specify an SA and request that the VMkernel use that SA.

The following options for SA setup are supported.

esxcli Option	Description
`sa-source <source_IP>`	Source IP for the SA.
`sa-destination <destination_IP>`	Destination IP for the SA.
`sa-spi`	Security Parameter Index (SPI) for the SA. Must be a hexadecimal number with a 0x prefix. When IPsec is in use, ESXi uses the ESP protocol (RFC 43030), which includes authentication and encryption information and the SPI. The SPI identifies the SA to use at the receiving host. Each SA you create must have a unique combination of source, destination, protocol, and SPI.
`sa-mode [tunnel | transport]`	Either tunnel or transport. In tunnel mode, the original packet is encapsulated in another IPv6 packet, where source and destination addresses are the SA endpoint addresses.
`encryption-algorithm [null | 3des-cbc | aes128-cbc]`	Encryption algorithm to be used. Choose `3des-cbc` or `aes128-cbc`, or `null` for no encryption.
`encryption-key <key>`	Encryption key to be used by the encryption algorithm. A series of hexadecimal digits with a 0x prefix or an ASCII string.

esxcli Option	Description
`integrity-algorithm [hmac-sha1 | hmac-sha2-256 ]`	Authentication algorithm to be used. Choose `hmac-sha1` or `hmac-sha2-256`.
`integrity-key`	Authentication key to be used. A series of hexadecimal digits or an ASCII string.

You can perform these main tasks with SAs.

- Create an SA. You specify the source, the destination, and the authentication mode. You also specify the authentication algorithm and authentication key to use. You must specify an encryption algorithm and key, but you can specify `null` if you want no encryption. Authentication is required and cannot be `null`. The following example includes extra line breaks for readability. The last option, `sa_2` in the example, is the name of the SA.

```
esxcli network ip ipsec sa add
            --sa-source 2001:DB8:1::121
            --sa-destination 2001:DB8:1::122
            --sa-mode transport
            --sa-spi 0x1000
            --encryption-algorithm 3des-cbc
            --encryption-key 0x697076367265616479 6c6f676f3364657363626f757432
            --integrity-algorithm hmac-sha1
            --integrity-key 0x69707636726561647 96c6f67736861316f757432
            --sa-name sa_2
```

- List an SA by using `esxcli network ip ipsec sa list`. This command returns SAs currently available for use by an SP. The list includes SAs you created.
- Remove a single SA by using `esxcli network ip ipsec sa remove`. If the SA is in use when you run this command, the command cannot perform the removal.
- Remove all SAs by using `esxcli network ip ipsec sa remove --removeall`. This option removes all SAs even when they are in use.

  Caution Running `esxcli network ip ipsec sa remove --removeall` removes all SAs on your system and might leave your system in an inconsistent state.

Managing Security Policies

After you have created one or more SAs, you can add security policies (SPs) to your ESXi hosts. While the SA specifies the authentication and encryption parameters to use, the SP identifies and selects traffic.

The following options for SP management are supported.

esxcli Option	Description
`sp-source <ip>/<p_len>`	Source IP address and prefix length.
`sp-destination <ip>/<p_len>`	Destination IP address and prefix length.
`source-port <port>`	Source port (0-65535). Specify `any` for any ports.

esxcli Option	Description
`destination-port <port>`	Destination port (0-65535). Specify `any` for any ports. If `ulproto` is `icmp6`, this number refers to the icmp6 type. Otherwise, this number refers to the port.
`upper-layer-protocol [any | tcp | udp | icmp6]`	Upper layer protocol. Use this option to restrict the SP to only certain protocols, or use any to apply the SP to all protocols.
`flow-direction [in | out]`	Direction in which you want to monitor the traffic. To monitor traffic in both directions, create two policies.
`action [none | discard | ipsec]`	Action to take when traffic with the specified parameters is encountered. ■ `none` - Take no action, that is, allow traffic unmodified. ■ `discard` - Do not allow data in or out. ■ `ipsec` - Use the authentication and encryption information specified in the SA to determine whether the data come from a trusted source.
`sp-mode [tunnel | transport]`	Mode, either `tunnel` or `transport`.
`sa-name`	Name of the SA to use by this SP.

You can perform the following main tasks with SPs.

- Create an SP by using `esxcli network ip ipsec add`. You identify the data to monitor by specifying the selector's source and destination IP address and prefix, source port and destination port, upper layer protocol, direction of traffic, action to take, and SP mode. The last two option are the name of the SA to use and the name of the SP that is being created. The following example includes extra line breaks for readability.

```
esxcli network ip ipsec add
   --sp-source=2001:0DB8:0001:/48
   --sp-destination=2001:0DB8:0002:/48
   --source-port=23
   --destination-port=25
   --upper-layer-protocol=tcp
   --flow-direction=out
   --action=ipsec
   --sp-mode=transport
   --sp-name sp_2
```

- List an SP by using `esxcli network ip ipsec list`. This command returns SPs currently available. All SPs are created by the administrator.
- Remove an SP by using `esxcli network ip ipsec remove`. If the SP is in use when you run this command, the command cannot perform the removal. You can run `esxcli network ip ipsec remove --removeall` instead to remove the SP even when it is in use.

  Caution Running `esxcli network ip ipsec remove --removeall` removes all SPs on your system and might leave your system in an inconsistent state.

Manage the ESXi Firewall

To minimize the risk of an attack through the management interface, ESXi includes a firewall between the management interface and the network.

To ensure the integrity of the host, only a small number of firewall ports are open by default. The *vSphere Security* documentation explains how to set up firewalls for your environment and which ports you might have to temporarily enable for certain traffic.

You manage firewalls by setting up firewall rulesets. The *vSphere Security* documentation explains how to perform these tasks by using the vSphere Client. You can also `use esxcli network firewall` to manage firewall rulesets and to retrieve information about them. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 Check firewall status and sshServer ruleset status.

```
esxcli <conn_options> network firewall get

Default Action: DROP
Enabled: true
Loaded: true

esxcli <conn_options> network firewall ruleset list --ruleset-id sshServer

Name       Enabled
---------  -------
sshServer     true
```

2 Enable the sshServer ruleset if it is disabled.

```
esxcli <conn_options> network firewall ruleset set --ruleset-id sshServer --enabled true
```

3 Obtain access to the ESXi Shell and check the status of the `allowedAll` flag.

```
esxcli <conn_options> network firewall ruleset allowedip list --ruleset-id sshServer

Ruleset    Allowed IP Addresses
---------  --------------------
sshServer  All
```

See *Getting Started with ESXCLI* for information on accessing the ESXi Shell.

4 Set the status of the `allowedAll` flag to `false`.

```
esxcli <conn_options> network firewall ruleset set --ruleset-id sshServer --allowed-all false
```

5 Add the list of allowed IP addresses.

```
esxcli <conn_options> network firewall ruleset allowedip add --ruleset-id sshServer --ip-address
192.XXX.1.0/24
esxcli <conn_options> network firewall ruleset allowedip add --ruleset-id sshServer --ip-address
192.XXX.10.10
```

6 Check the allowed IP address list.

```
esxcli <conn_options> network firewall ruleset allowedip list --ruleset-id sshServer

Ruleset    Allowed IP Addresses
---------  ------------------------------
sshServer  192.XXX.10.10, 192.XXX.1.0/24
```

Monitor VXLAN

The `esxcli network vswithch dvs vmware vxlan` namespace supports commands for exploring VXLAN configuration details.

For a more detailed example of this functionality, see the VMware vSphere blog post about the topic.

Procedure

1 List all available VXLAN vNetwork Distributed Switches.

```
esxcli network vswitch dvs vmware vxlan list
```

2 View the VXLAN statistics level.

```
esxcli network vswitch dvs vmware vxlan config stats get
```

3 Change the statistics level, for example, from 0 to 1.

```
esxcli network vswitch dvs vmware vxlan config stats set --level 1
```

You can decide to filter statistics as follows.

- For a vNetwork Distributed Switch, localized to an ESXi host
- For a VTEP VMkernel interface
- For a VXLAN segment ID
- For a vNetwork Distributed Switch port ID

4 View statistics for a specific vNetwork Distributed Switch.

```
esxcli network vswitch dvs vmware vxlan config stats list --vds-name Cluster01-VXLAN-VDS
```

5 View statistics for a VXLAN segment ID.

- List the available segment IDs.

```
esxcli network vswitch dvs vmware vxlan network list –vds–name Cluster01–VXLAN–VDS
```

- View the network statistics for a particular segment ID.

```
esxcli network vswitch dvs vmware vxlan network stats list ––vds–name Cluster01–VXLAN–VDS ––
vxlan–id 5000
```

- Retrieve network mapping if some virtual machine communication is occurring.

```
esxcli network vswitch dvs vmware vxlan network mapping list ––vds–name Cluster01–VXLAN–VDS ––
vxlan–id 5000
```

6 View VXLAN statistics for a VDS Port ID.

```
esxcli network vswitch dvs vmware vxlan network port list ––vds–name Cluster01–VXLAN–VDS ––vxlan–
id 5000
```

7 View the network statistics for a specific VDS Port ID.

```
esxcli network vswitch dvs vmware vxlan network port list ––vds–name Cluster01–VXLAN–VDS ––vxlan–
id 5000 vdsport–is 968
```

Monitoring ESXi Hosts

10

vCenter Server makes performance charts for CPU, memory, disk I/O, networking, and storage available.

You can view performance charts by using the vSphere Client and read about them in the *vSphere Monitoring* documentation. You can also perform some monitoring of your ESXi system by using ESXCLI commands.

This chapter includes the following topics:

Managing Diagnostic Partitions

Your host must have a diagnostic partition, also referred to as dump partition, to store core dumps for debugging and for use by VMware technical support.

A diagnostic partition is on the local disk where the ESXi software is installed by default. You can also use a diagnostic partition on a remote disk shared between multiple hosts. If you want to use a network diagnostic partition, you can install ESXi Dump Collector and configure the networked partition. See Manage Core Dumps with ESXi Dump Collector.

The following considerations apply.

- A diagnostic partition cannot be located on an iSCSI LUN accessed through the software iSCSI or dependent hardware iSCSI adapter. For more information about diagnostic partitions with iSCSI, see General Boot from iSCSI SAN Recommendations in the *vSphere Storage* documentation.
- A standalone host must have a diagnostic partition of 110 MB.
- If multiple hosts share a diagnostic partition on a SAN LUN, configure a large diagnostic partition that the hosts share.

- If a host that uses a shared diagnostic partition fails, reboot the host and extract log files immediately after the failure. Otherwise, the second host that fails before you collect the diagnostic data of the first host might not be able to save the core dump.

Diagnostic Partition Creation

You can use the vSphere Client to create the diagnostic partition on a local disk or on a private or shared SAN LUN. The SAN LUN can be set up with FibreChannel or hardware iSCSI. SAN LUNs accessed through a software iSCSI initiator are not supported.

Caution If two hosts that share a diagnostic partition fail and save core dumps to the same slot, the core dumps might be lost.

If a host that uses a shared diagnostic partition fails, reboot the host and extract log files immediately after the failure.

Diagnostic Partition Management

You can use the `esxcli system coredump` command to query, set, and scan an ESXi system's diagnostic partitions. The *vSphere Storage* documentation explains how to set up diagnostic partitions with the vSphere Client and how to manage diagnostic partitions on a Fibre Channel or hardware iSCSI SAN.

Diagnostic partitions can include, in order of suitability, parallel adapter, block adapter, FC, or hardware iSCSI partitions. Parallel adapter partitions are most suitable and hardware iSCSI partitions the least suitable.

Important When you list diagnostic partitions, software iSCSI partitions are included. However, SAN LUNs accessed through a software iSCSI initiator are not supported as diagnostic partitions.

Managing Core Dumps

With `esxcli system coredump`, you can manage local diagnostic partitions or set up core dump on a remote server in conjunction with the ESXi Dump Collector.

For information about the ESXi Dump Collector, see the *vSphere Networking* documentation.

Manage Local Core Dumps with ESXCLI

You can use ESXCLI to manage local core dumps.

The following example scenario changes the local diagnostic partition by using ESXCLI. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1. Show the diagnostic partition the VMkernel uses and display information about all partitions that can be used as diagnostic partitions.

   ```
   esxcli <conn_options> system coredump partition list
   ```

2. Deactivate the current diagnostic partition.

   ```
   esxcli <conn_options> system coredump partition set --unconfigure
   ```

   The ESXi system is now without a diagnostic partition, and you must immediately set a new one.

3. Set the active partition to `naa.<naa_ID>`.

   ```
   esxcli <conn_options> system coredump partition set --partition=naa.<naa_ID>
   ```

4. List partitions again to verify that a diagnostic partition is set.

   ```
   esxcli <conn_options> system coredump partition list
   ```

   If a diagnostic partition is set, the command displays information about it. Otherwise, the command shows that no partition is activated and configured.

Manage Core Dumps with ESXi Dump Collector

By default, a core dump is saved to the local disk. You can use the ESXi Dump Collector to keep core dumps on a network server for use during debugging.

The ESXi Dump Collector is especially useful for Auto Deploy. The ESXi Dump Collector supports other customization, including sending core dumps to the local disk.

The ESXi Dump Collector is included with the vCenter Server `autorun.exe` application. You can install the ESXi Dump Collector on the same system as the vCenter Server service or on a different Windows or Linux machine. See *vSphere Networking*.

You can configure ESXi hosts to use the ESXi Dump Collector by using the Host Profiles interface of the vSphere Client, or by using ESXCLI. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1. Set up an ESXi system to use the ESXi Dump Collector by running `esxcli system coredump`.

   ```
   esxcli <conn_options> system coredump network set --interface-name vmk0 --server-ipv4=1-XX.XXX --
   server-port=6500
   ```

   You must specify a VMkernel port with `--interface-name`, and the IP address and port of the server to send the core dumps to. If you configure an ESXi system that is running inside a virtual machine, you must choose a VMkernel port that is in promiscuous mode.

2 Enable the ESXi Dump Collector.

```
esxcli <conn_options> system coredump network set --enable=true
```

3 (Optional) Check that the ESXi Dump Collector is configured correctly.

```
esxcli <conn_options> system coredump network get
```

Results

The host on which you have set up the ESXi Dump Collector sends core dumps to the specified server by using the specified VMkernel NIC and optional port.

Configuring ESXi Syslog Services

All ESXi hosts run a syslog service, which logs messages from the VMkernel and other system components to local files or to a remote host.

You can use the vSphere Client, or use the `esxcli system syslog` command to configure the following parameters of the syslog service.

- Remote host and port - Remote host to which syslog messages are forwarded and port on which the remote host receives syslog messages. The remote host must have a log listener service installed and correctly configured to receive the forwarded syslog messages. See the documentation for the syslog service installed on the remote host for information on configuration.
- Transport protocol - Logs can be sent by using UDP, which is the default, TCP, or SSL transports.
- Local logging directory - Directory where local copies of the logs are stored. The directory can be located on mounted NFS or VMFS volumes. Only the `/scratch` directory on the local file system is persistent across reboots.
- Unique directory name prefix - Setting this option to true creates a subdirectory with the name of the ESXi host under the specified logging directory. This method is especially useful if the same NFS directory is used by multiple ESXi hosts.
- Log rotation policies - Sets maximum log size and the number of archives to keep. You can specify policies both globally, and for individual subloggers. For example, you can set a larger size limit for the `vmkernel` log.

After making configuration changes, restart the `vmsyslogd` syslog service by running `esxcli system syslog reload`.

The `esxcli system syslog` command allows you to configure the logging behavior of your ESXi system. You can manage the top-level logger and subloggers. The command has the following options.

Option	Description
`mark`	Marks all logs with the specified string.
`reload`	Reloads the configuration, and updates any changed configuration values.
`config get`	Retrieves the current configuration.

Option	Description
`config set`	Sets the configuration. Use one of the following options. ■ `--logdir=<path>` – Saves logs to a given path. ■ `--loghost=<host>` – Sends logs to a given host. ■ `--logdir-unique=<true|false>` – Specifies whether the log should go to a unique subdirectory of the directory specified in `logdir`. ■ `--default-rotate=<int>` – Default number of log rotations to keep. ■ `--default-size=<int>` – Size before rotating logs, in KB.
`config logger list`	Shows currently configured subloggers.
`config logger set`	Sets configuration options for a specific sublogger. Use one of the following options. ■ `--id=<str>` – ID of the logger to configure. Required. ■ `--reset=<str>` – Resets values to default. ■ `--rotate=<long>` – Number of rotated logs to keep for a specific logger. Requires `--id`. ■ `--size=<long>` – Size of logs before rotation for a specific logger, in KB. Requires `--id`.

Example: esxcli system syslog Usage

The following workflow illustrates how you might use `esxcli system syslog` for log configuration. Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

1 Show configuration options.

```
esxcli <conn_options> system syslog config get
Default Rotation Size: 1024
Default Rotations: 8
Log Output: /scratch/log
Logto Unique Subdirectory: false
Remote Host: <none>
```

2 Set all logs to keep twenty rotations before overwriting the oldest log.

```
esxcli <conn_options> system syslog config set --default-rotate=20
```

3 Set the rotation policy for VMkernel logs to 10 rotations, rotating at 2 MB.

```
esxcli <conn_options> system syslog config logger --id=vmkernel --size=2048 --rotate=10
```

4 Send logs to remote host `myhost.mycompany.com`. The logs will use the default transport (UDP) and port (514).

```
esxcli system syslog config set --loghost='myhost.mycompany.com'
```

5 Save the local copy of logs to `/scratch/mylogs` and send another copy to the remote host.

```
esxcli <conn_options> system syslog config set --loghost='tcp://myhost.mycompany.com:1514' --
logdir='/scratch/mylogs'
```

You can set the directory on the remote host by configuring the client running on that host. You can use the vSphere Client to redirect system logs to a remote host by changing the `System.global.logHost` advanced setting.

6 Send a log message to all logs simultaneously.

```
esxcli <conn_options> system syslog mark ––message="this is a message!"
```

7 Reload the syslog daemon and apply configuration changes.

```
esxcli <conn_options> system syslog reload
```

Managing ESXi SNMP Agents

Simple Network Management Protocol (SNMP) allows management programs to monitor and control networked devices.

The host-based embedded SNMP agent is disabled by default. Configuring and enabling the agent requires that you perform the following tasks.

1 Configure SNMP communities. See Configuring SNMP Communities.

2 Configure the SNMP agent. See Configuring the SNMP Agent to Send Traps.

Configuring SNMP Communities

Before you enable the ESXi embedded SNMP agent, you must configure at least one community for the agent.

An SNMP community defines a group of devices and management systems. Only devices and management systems that are members of the same community can exchange SNMP messages. A device or management system can be a member of multiple communities.

To configure SNMP communities, run `esxcli system snmp set`, specifying a comma-separated list of communities as shown in the following example.

```
esxcli system snmp set –c public, internal
```

Each time you specify a community with this command, the settings that you specify overwrite the previous configuration.

Configuring the SNMP Agent to Send Traps

You can use the SNMP agent embedded in ESXi to send virtual machine and environmental traps to management systems.

To configure the agent to send traps, you must specify a target address, also referred to as receiver address, the community, and an optional port. If you do not specify a port, the SNMP agent sends traps to UDP port 162 on the target management system by default.

Configure a Trap Destination with ESXCLI

You can use ESXCLI to configure a trap destination and send traps.

Specify one of the options listed in Connection Options for ESXCLI Host Management Commands in place of `<conn_options>`.

Procedure

1 Make sure a community is set up.

```
esxcli system snmp get <conn_options>

Current SNMP agent settings:
Enabled: 1
UDP port: 161
Communities: public
Notification targets:
```

2 Set the target address, port number, and community.

```
esxcli <conn_options> system snmp set –t target.example.com@163/public
```

Each time you specify a target with this command, the settings you specify overwrite all previously specified settings. To specify multiple targets, separate them with a comma.

You can change the port that the SNMP agent sends data to on the target using the `––targets` option. That port is UDP 162 by default.

3 (Optional) Enable the SNMP agent if it is not yet running.

```
esxcli <conn_options> system snmp set ––enable=yes
```

4 (Optional) Send a test trap to verify that the agent is configured correctly.

```
esxcli <conn_options> system snmp test
```

The agent sends a `warmStart` trap to the configured target.

Retrieving Hardware Information

Commands in different ESXCLI namespaces might display some hardware information, but the `esxcli hardware` namespace is specifically intended to give you that information. The namespace includes commands for getting and setting CPU properties, for listing boot devices, and for getting and setting the hardware clock time.

You can also use the `ipmi` namespace to retrieve IPMI system event logs (SEL) and sensor data records (SDR). The command supports both `get` (single return value) and `list` (multiple return values) commands and returns raw sensor information.

See the *ESXCLI Reference* for details.