iSCSI Design Considerations and Deployment Guide

VMware Infrastructure 3

With the release of VMware Infrastructure 3, VMware added ESX Server support for iSCSI storage. With rapidly increasing adoption rates, many VMware customers requested iSCSI as an option for storage virtualization and are now deploying it as a lower cost alternative to Fibre Channel SANs.

This paper is intended to help you understand the design considerations and deployment options for deploying VMware Infrastructure 3 using iSCSI storage. The first section provides an overview of iSCSI terminology, benefits, and limitations. The second section provides a high-level overview of the VMware iSCSI implementation using either a software initiator or a hardware initiator. The third section provides a detailed set of deployment steps covering both software and hardware initiator options. The paper concludes with two appendices that provide software versus hardware initiator iSCSI performance test results and details on command line options for managing iSCSI from the ESX Server host.

This paper highlights trade-offs and factors to consider when deploying iSCSI storage to support VMware Infrastructure 3. It is a complement to, not a replacement for, VMware product documentation.

iSCSI Overview

iSCSI is a protocol that uses TCP to transport SCSI commands for a storage network, enabling existing TCP/IP infrastructure to be used as a SAN. As with SCSI over Fibre Channel, iSCSI presents SCSI LUNs (targets) to iSCSI initiators (requesters). Unlike NAS, which presents devices at the file level, iSCSI makes block devices available via the network. You can therefore mount block devices (disks) across an IP network to your local system, then use them as you would any other block device.

iSCSI Benefits

Data centers are rapidly moving from distributed to centralized storage. iSCSI offers customer many benefits because it is relatively inexpensive and based on familiar SCSI and TCP/IP standards. In comparison to Fibre Channel SAN deployments, iSCSI requires less hardware, it uses lower-cost hardware, and more IT staff members are familiar with the technology. These factors contribute to lower-cost implementations.

iSCSI Drawbacks

One major reason that customers may not deploy iSCSI is that Fibre Channel SANs represent the more established and mature technology in the storage world. iSCSI is still relatively new. It is still advancing in stability and performance as it becomes more widely deployed, but it does not yet generally provide performance as high as Fibre Channel SANs. A basic difference between iSCSI and Fibre Channel is that when an iSCSI path is overloaded, the TCP/IP protocol drops packets and requires them to be resent. Fibre Channel communications over a dedicated path are not at risk of being overloaded. When a network path carrying iSCSI storage traffic is oversubscribed, a bad situation quickly grows worse and performance further degrades as dropped packets must be resent.

Another potential disadvantage with implementing software-initiator iSCSI (but not hardware-initiator iSCSI) is that standard 10/100 Ethernet interfaces do not have enough throughput for practical iSCSI work. Gigabit Ethernet interfaces are required, and those interfaces tend to consume large amounts of CPU time. One way of overcoming this demand for CPU resources is to use TOEs (TCP/IP offload engines). TOEs shift TCP packet processing tasks from the server CPU to specialized TCP processors on the network adapter or storage device. The QLA4050 hardware initiator, which is supported on ESX Server 3.0, uses a TOE.

Finally, iSCSI does not work well over most shared wide area networks. For a more detailed discussion of iSCSI over WANs, see "iSCSI Design Considerations" on page 8.

iSCSI Architecture

iSCSI initiators must manage multiple, parallel communications links to multiple targets. Similarly, iSCSI targets must manage multiple, parallel communications links to multiple initiators. Several identifiers exist in iSCSI to make this happen:

- iSCSI names
- ISIDs (iSCSI session identifiers) and TSID (target session ID)
- CIDs (iSCSI connection identifiers)
- iSCSI portals

iSCSI Names Are Globally Unique

iSCSI nodes have globally unique names similar to the World Wide Names (WWN) used in a SAN environment. iSCSI names do not change when Ethernet adapters or IP addresses change. iSCSI supports EUI and IQN names as well as aliases. EUI (extended unique identifier) is the IEEE EUI-64 format. IQN is the iSCSI qualified name.

Both EUI-48 and EUI-64 define the first 24 bits as the Company_ID, which is administered by the IEEE Registration Authority Committee (http://standards.ieee.org/regauth).

Example of an IQN name:

```
iqn.1984-08.com.whatzis:hedgetrimmer-1926184
```

Example of an EUI name:

```
eui.02004567A425678D
```

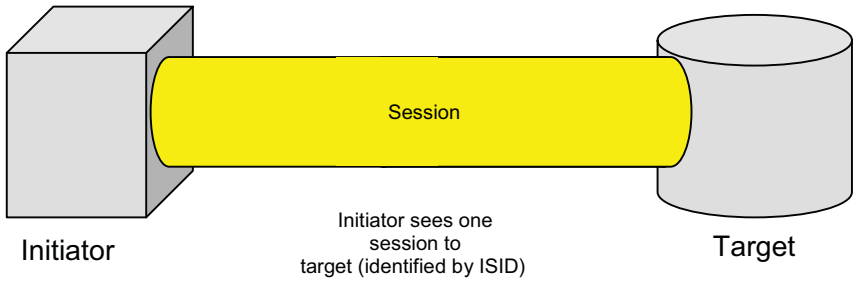
iSCSI Initiators and Targets Are the End Points

A storage network consists of two types of equipment: initiators and targets. Initiators are data consumers, such as hosts. Targets are data providers, such as disk arrays or tape libraries. Initiators and targets, collectively referred to as end points, can be software, software with hardware assistance, or hardware. This section examines the features and issues with each of these technologies.

iSCSI Sessions and Connections Create the Paths

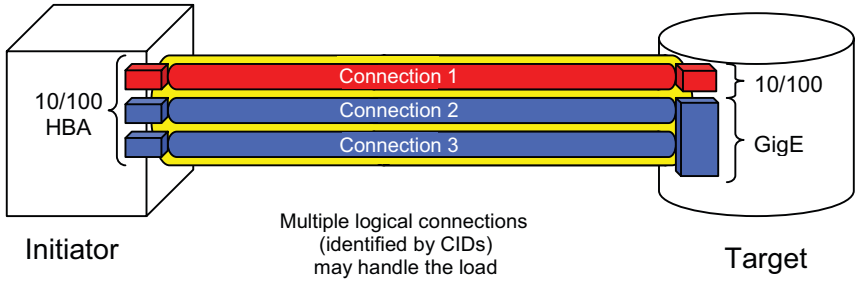
iSCSI nodes use TCP to create relationships called sessions. Session IDs (ISIDs) are not tied to hardware and can persist across hardware swaps. The initiator sees one session to the target (identified by ISID), as shown in Figure 1.

Figure 1. iSCSI session identified by ISID



An iSCSI session may create multiple logical connections. Connections aggregate bandwidth and provide load balancing. Multiple logical connections to the target (identified by CIDs) may handle the load, as shown in Figure 2. VMware Infrastructure 3 does not currently support multiple TCP session connections (MSC).

Figure 2. iSCSI logical connections identified by CIDs

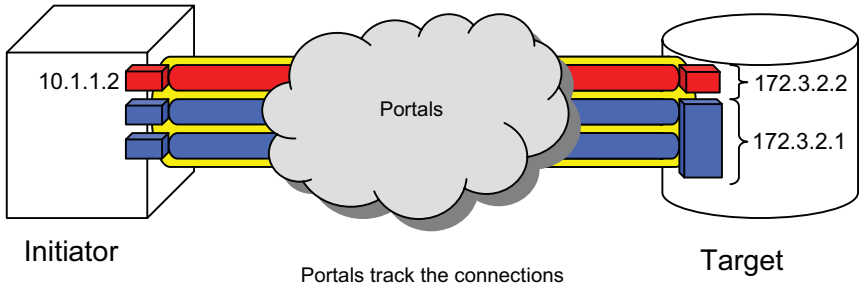


iSCSI Portals Keep Track of Connections

iSCSI nodes keep track of connections via portals, which allow separation between names and IP addresses. A portal manages an IP address and a TCP port number. Therefore, from an architectural perspective, sessions can be made up of multiple logical connections, and portals track connections via TCP/IP port/address, as shown in Figure 3.

However, because VMware Infrastructure 3 does not support a multiple connections per session, it should have only a single portal tracking a connection.

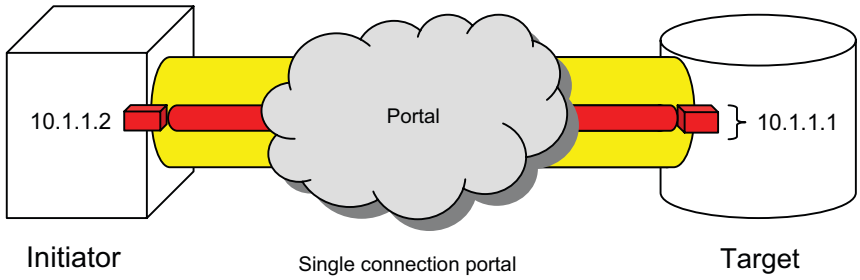
Figure 3. iSCSI portal tracks multiple connections



iSCSI Driver Implementation in VMware Infrastructure 3

The iSCSI driver used by ESX Server does not currently use multiple connections per session, so there are no such settings that you can tune. The ESX Server 3 iSCSI configuration works as shown in Figure 4

Figure 4. VMware Infrastructure 3 iSCSI configuration



iSCSI Security

The most recent version of the iSCSI protocol supports several types of security:

Encryption

- IPSec (Internet Protocol security) is a developing standard for security at the network or packet processing layer of network communication.
- IKE (Internet key exchange) is an IPsec standard protocol used to ensure security for VPNs.

Authentication

- Kerberos v5.1—not widely implemented
- SRP (Secure Remote Password)—not widely implemented
- SPKM1/2 (simple public-key mechanism)—not widely implemented
- CHAP (Challenge Handshake Authentication Protocol)

Challenge Handshake Authentication Protocol

CHAP verifies identity using a hashed transmission. The target initiates the challenge. The secret key is known by both parties. It periodically repeats the challenge to guard against replay attacks. CHAP is a one-way protocol, but it may be implemented in two directions to provide security for both ends.

Because the current version of the iSCSI specification defines the CHAP security method as the only must-support protocol, the VMware implementation uses this security option. However, bidirectional CHAP is not currently supported on ESX Server 3.

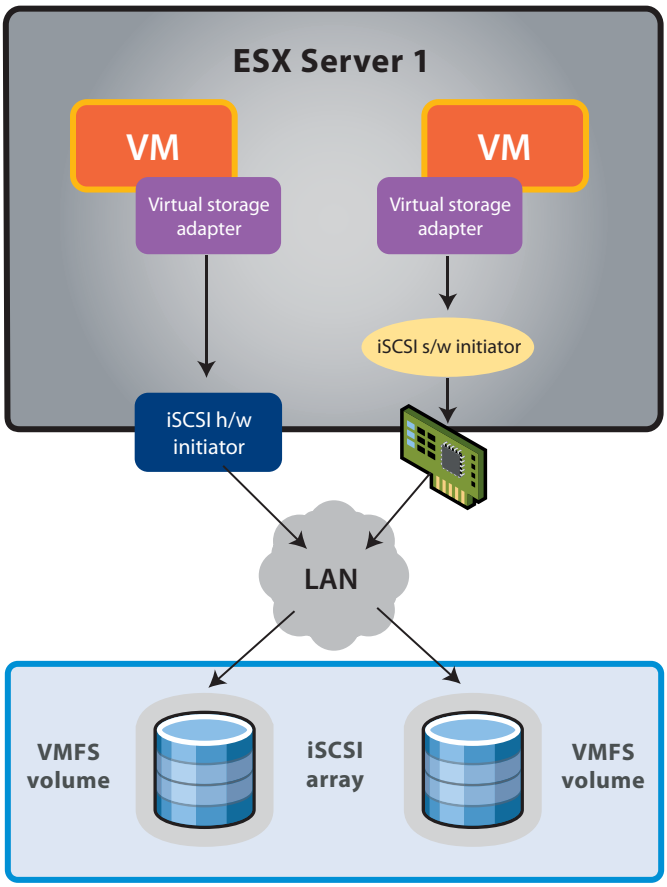
Jumbo Frames

In a high volume transmission environment use of larger packet size can reduce the overhead of processing Ethernet packets and increase performance. A jumbo frame is an Ethernet frame that carries more than the standard 1500 byte payload. Typically a jumbo frame is 9000 bytes, but the size can vary. Testing has shown increases performance on the order of 30 percent due to the reduced assembly and disassembly packet processing. However, for jumbo frames to be used, both end points and all devices (switches and routers) in between those end points must have jumbo frames enabled. At present, the VMware software initiator does not support jumbo frames. And until 10 gigabit Ethernet is supported by the VMware software initiator, the performance benefit of using jumbo frames would be minimal. Support for both jumbo frames and 10 gigabit Ethernet are planned for a future release of VMware Infrastructure 3.

VMware ESX Server 3 iSCSI Implementation

With the release of ESX Server 3.0, VMware added support for iSCSI with both software initiator and hardware initiator implementations. The software initiator iSCSI plugs into the ESX Server storage stack as a device driver similar to other SCSI and Fibre Channel drivers. This means that it implicitly supports VMFS-3, RDM, and raw device access. ESX Server 3.0.x supports software iSCSI and hardware iSCSI initiators. Booting from iSCSI is supported for hardware iSCSI only. Figure 5 shows the differences between an iSCSI hardware and iSCSI software implementation in ESX Server 3.

Figure 5. iSCSI hardware and software initiators



Choosing an iSCSI Option

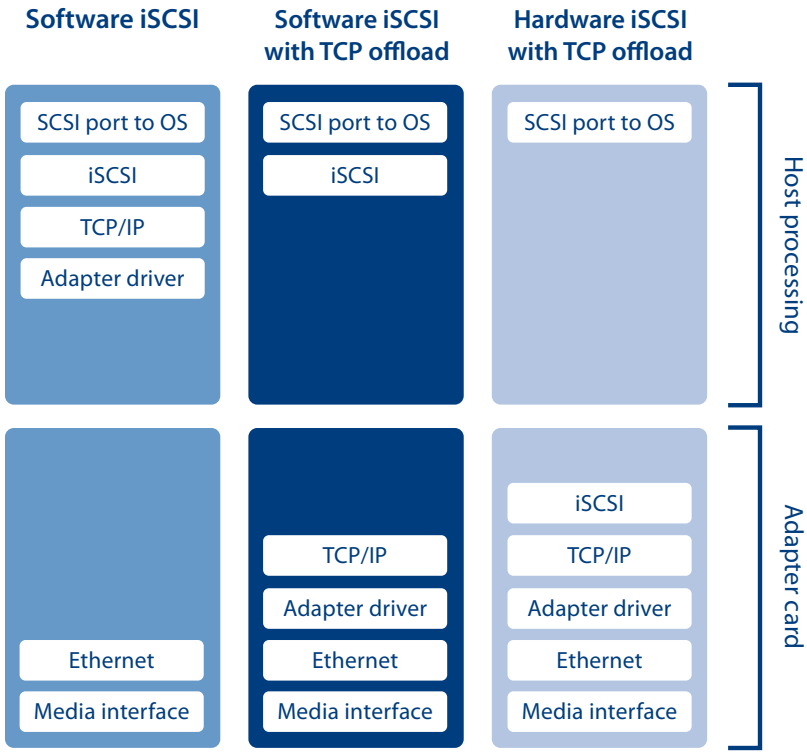
With ESX Server 3, you can use either the software initiator or the hardware initiator. Use of both initiators on the same ESX Server host is not supported.

With the hardware-initiator iSCSI implementation, the iSCSI HBA provides the translation from SCSI commands to IP on the adapter. ESX Server sees the adapter as a Fibre Channel HBA device. The network sees the adapter as a network adapter. This hardware device is referred to as a TCP offload engine adapter. It offloads the process of translation from the host server and under some workloads frees the sever CPU from doing some extra work. This can improve server performance if CPU resources are significantly taxed. For hardware initiators, there is currently a limit of two ports per ESX Server host—either one dual-port iSCSI HBA or two single-port iSCSI HBAs.

The software-initiator iSCSI implementation leverages the VMkernel to perform the SCSI to IP translation and does require extra CPU cycles to perform this work. As a result, software iSCSI can reduce overall server performance when CPUs are under heavy load. The software and hardware implementations do not currently support jumbo frames, multisession connect or 10gigabit Ethernet speed. Additional considerations for deciding which initiator to use are covered in "Current VMware Infrastructure 3 iSCSI Support Status" on page 7.

Figure 6 shows the VMware hardware-initiator iSCSI implementation (iSCSI HBA) on the right and the software-initiator implementation on the left. Between them is a hybrid version of a TOE adapter that is on the market today but is not currently supported by ESX Server.

Figure 6. Software and hardware iSCSI implementations (source: *iSCSI Building Blocks for IP Storage Networking* on http://www.snia.org)



Software iSCSI

The iSCSI protocol for block storage I/O is based on standard Ethernet transports, including Fast Ethernet (100Mbps) and Gigabit Ethernet (1.25Gbps). As a very economical, low-performance block I/O solution, iSCSI drivers can be run on an off-the-shelf Ethernet adapter. For example, a server or workstation with an existing Gigabit Ethernet adapter can run iSCSI in software to perform block I/O tape backup over an IP network. This is more efficient than traditional file-based backup and requires less than 20Mbps of bandwidth.

The cost advantage of software iSCSI must be balanced against the penalty it can impose on performance and the CPU overhead it requires. When there is no embedded TCP offload engine, the CPU handles TCP processing. In addition, the host CPU has to process the iSCSI protocol that is carried by TCP/IP. iSCSI software solutions may be suitable for medium-performance storage applications such as tape backup. They also offer a new mid-range option for storage networking unavailable in Fibre Channel SANs. For gigahertz and higher-speed server CPUs, the overhead of TCP and iSCSI processing may be an acceptable trade-off for the convenience of iSCSI storage access.

Software iSCSI with TCP Offload

The development of TCP offload engines marks a significant advance for all TCP/IP based protocols, including iSCSI. Although TCP provides session integrity in potentially congested or unstable networks, it incurs significant processing overhead on the host CPU. Offloading this processing to a host Ethernet adapter frees host CPU cycles and enables much higher performance solutions.

Because block I/O transactions can generate sustained high volumes of TCP/IP exchanges, IP storage is a direct beneficiary of TOE technology. With the burden of TCP overhead removed from the host CPU, only iSCSI processing is required. The remaining challenge for CPU utilization is optimizing iSCSI handling so blocks of data can be served to the host more efficiently. With clever engineering, wire-speed performance can be achieved when running software iSCSI on an optimized TOE-accelerated network adapter.

Hardware iSCSI with TCP Offload (iSCSI HBA)

iSCSI HBAs are interface adapters that provide high-performance offloading of both TCP and iSCSI processing onto the adapter. Although this adds cost to the adapter, it delivers high-speed iSCSI transport with minimal CPU overhead. The iSCSI processing may be performed in a custom ASIC or in an onboard processor. In either case, the SCSI commands, status, and data encapsulated by iSCSI are served up by the adapter directly to the host operating system.

Current VMware Infrastructure 3 iSCSI Support Status

At the time of publication, VMware supports QLA401x and QLA405x for use as iSCSI HBAs and is investigating iSCSI HBAs from additional vendors for future support. For software iSCSI, VMware supports any network adapter listed in the hardware compatibility list, and additional cards are being evaluated as market demand warrants. VMware does not, however, currently support any iSCSI software with TOE cards. So far, VMware has found that limited performance gains are provided by these cards in comparison to software-only iSCSI via network adapters. Support for iSCSI software with TOE cards is under consideration and may change in the future. It is also likely to become more of an option as jumbo frame support and the faster interconnect speeds of 10 gigabit Ethernet are adopted more widely in the industry.

Hardware iSCSI Features and Limitations

Below are some items to consider if you are planning to use the iSCSI hardware intitiator.

- ESX Server host booting from iSCSI SAN is possible only with hardware iSCSI initiator
- Multipathing support for failover only, no load-balancing by using multiple QLA4010s
- Support for VMotion, VMware HA, and VMware DRS
- Support for RDMs
- No support for Microsoft Cluster Server
- No VMware Consolidated Backup over iSCSI

Software iSCSI Features and Limitations

Below are some items to consider if you are planning to use the iSCSI software intitiator.

- No support for booting ESX Server from software iSCSI
- Software initiator supports only a single storage interface (vmhba40)
- Multipathing support for failover only, no load balancing by using multiple physical network adapters (NIC teaming)
- Support for VMotion, VMware HA, and VMware DRS
- Support for RDMs
- No support for Microsoft Cluster Server
- No VMware Consolidated Backup over iSCSI

The decision between hardware and software initiators is essentially a trade-off between price and performance. With network adapters costing about $100 each and iSCSI HBAs costing about $550, the difference of $450 saved needs to be weighed against the impact on both throughput and CPU utilization. As a sample test of that impact, VMware conducted a simple test using IOMeter as the workload generator to measure differences between the two approaches with a heavy workload. While the same workload was run on identical servers and isolated storage resources, the relative throughput for the iSCSI software initiator was lower than that for the hardware initiator, and the average CPU utilization using the iSCSI software initiator was higher than that using the hardware initiator. The hardware initiator delivered about 150 percent of the

throughput of the software initiator while requiring 25 percent of the CPU resources used by the software initiator.

NOTE If you plan to use NIC teaming to increase the availability of your network access to the iSCSI storage array, you must turn off port security on the switch for the two ports on which the virtual IP address is shared. The purpose of this port security setting is to prevent spoofing of IP addresses. Thus many network administrators enable this setting. However, if you do not change it, the port security setting prevents failover of the virtual IP from one switch port to another and NIC teaming cannot fail over from one path to another. For most LAN switches, the port security is enabled on a port level and thus can be set on or off for each port.

iSCSI Design Considerations

Network design is key to making sure iSCSI works. In a production environment, Gigabit Ethernet is essential for software iSCSI. Hardware iSCSI, in a VMware Infrastructure environment, is implemented with dedicated QLogic QLA4010 and 405x HBAs.

You should consider iSCSI a local-area technology, not a wide-area technology, because of latency issues and security concerns. You should also segregate iSCSI traffic from general traffic. Layer 2 VLANs are a particularly good way to implement this segregation.

Beware of oversubscription. Oversubscription refers to connecting more users to a system than can be fully supported if all of them are using it at the same time. Networks and servers are almost always designed with some amount of oversubscription, assuming that users do not all need the service simultaneously. If they do, delays are certain and outages are possible. Oversubscription is permissible on general-purpose LANs, but you should not use an oversubscribed configuration for iSCSI.

Best practice is to have a dedicated LAN for iSCSI traffic and not share the network with other network traffic. It is also best practice not to oversubscribe the dedicated LAN.

A Single VMkernel Gateway

The VMkernel has a single routing table for all its VMkernel Ethernet interfaces. This imposes some limits on network communication. Consider a configuration that uses two Ethernet adapters with one VMkernel TCP/IP stack. One adapter is on the 10.17.1.1/24 IP network and the other on the 192.168.1.1/24 network. Assume that 10.17.1.253 is the address of the default gateway. The VMkernel can communicate with any servers reachable by routers that use the 10.17.1.253 gateway. It may not be able to talk to all servers on the 192.168 network unless both networks are on the same broadcast domain.

The VMkernel TCP/IP Routing Table

Another consequence of the single routing table affects one approach you might otherwise consider for balancing I/O. Consider a configuration in which you want to connect to iSCSI storage and also want to allow NFS mounts. It might seem that you could use one Ethernet adapter for iSCSI and a separate Ethernet adapter for NFS to spread the I/O load. This approach does not work because of the way the VMkernel TCP/IP stack handles entries in the routing table.

For example, you might assign an IP address of 10.16.156.66 to the adapter you want to use for iSCSI. The routing table then contains an entry for the 10.16.156.x network for this adapter. If you then set up a second adapter for NFS and assign it an IP address of 10.16.156.25, the routing table contains a new entry for the 10.16.156.x network for the second adapter. However, when the TCP/IP stack reads the routing table, it never reaches the second entry, because the first entry satisfies all routes to both adapters. Therefore, no traffic ever goes out on the NFS network.

The fact that all 10.16.156.x traffic is routed on the iSCSI network causes two types of problems. First, you do not see any traffic on the second Ethernet adapter. Second, if you try to add trusted IP addresses to both iSCSI arrays and NFS servers, traffic to one or the other comes from the wrong IP address. For example, if the NDFS server `/etc/exports` file is set up to accept connections from 10.16.156.25 but requests to mount come from 10.16.156.66, the requests are not honored.

iSCSI Deployments Steps

Before configuring an ESX Server host to address storage via iSCSI, be sure you have taken the appropriate preliminary steps to set up your storage array and ensure that communication is working from both ends of the connection. Each storage vendor has its own set of steps to make LUNs available on the target initiator. You may need to consult your storage administrator or vendor documentation to complete the following steps.

1 Make sure the storage resource is connected to the network, that you know its IP address, and that it responds to a network ping at that address.

2 Provision one or more LUNs to the network port on the iSCSI storage device.

3 Note the IQN (iSCSI qualified name) for the storage device.

4 The IQN of the storage device (target) side of the connection must be discovered by the initiator once it is configured on the ESX Server host.

5 Make the following configuration settings on the ESX Server host:

- Hardware or software initiator
- IP address for the ESX Server end of the iSCSI link
- Discovery method—static configuration or send targets
- Use CHAP authentication or not—default is to use CHAP

6 If you are using a hardware initiator, install the device in the ESX Server host and follow the steps in "Configuring a Hardware iSCSI Initiator" on page 9. If you are using the software initiator, skip to "Configuring a Software iSCSI Initiator" on page 11.

Configuring a Hardware iSCSI Initiator

To ESX Server, a hardware initiator looks like any other SCSI adapter. SCSI LUNs are made available to ESX Server from the iSCSI adapter. The iSCSI adapter has its own networking stack, which is referred to as the TOE (the TCP offload engine). As a result, the guest operating system does not handle any of the networking required for access to the iSCSI storage.

To configure hardware iSCSI— in other words, to configuring the HBAs— take the following steps:

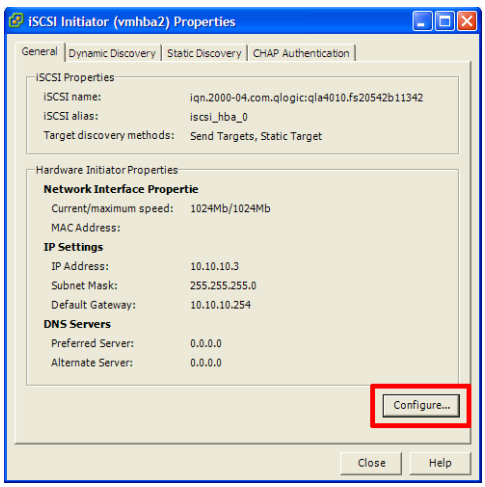
1 Start the VI Client and click the configuration tab for the ESX Server host you want to configure.

2 Open the Storage Adapters panel.

Choose **Configuration** > **Storage Adapters**.

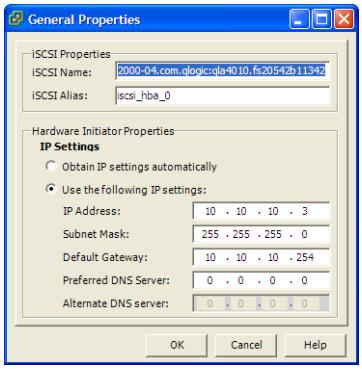
3 Right-click the HBA you want to configure, then choose **Properties**.

The iSCSI Initiator Properties dialog box appears.



4 Click **Configure**.

The General Properties dialog box appears.



5 Set the IP information for this HBA. You can make the following changes:

- Change the **iSCSI Name** if you do not want to use the default.
- Enter a value for **iSCSI Alias**.
- Set the IP information for the hardware initiator.

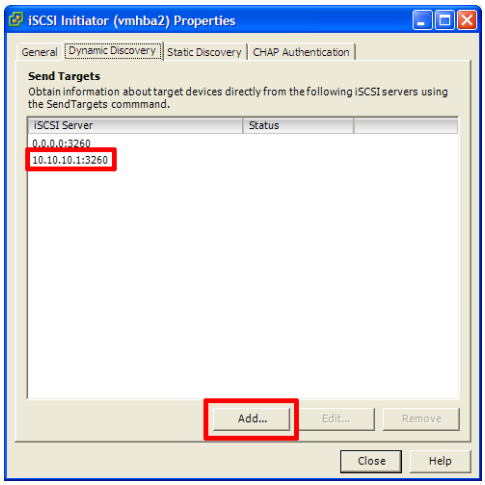
NOTE It is important to set an IP address for the iSCSI HBA that is different from the IP address for the ESX Server host. If they are the same, severe networking problems occur and you may be unable to communicate with the ESX Server host.

When you have made all the changes you want to make, click **OK**.

The iSCSI Initiator Properties dialog box appears again.

6 Click the **Dynamic Discovery** tab.

The Dynamic Discovery dialog box appears.



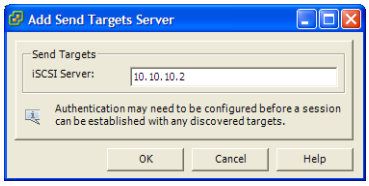
In this dialog box, you can add the IP addresses of the iSCSI array (target) ports that the hardware initiator can discover. The iSCSI HBA must be in the same VLAN as both ports of the iSCSI array.

In the example used in this paper, the IP addresses of the iSCSI ports are 10.10.10.1 and 10.10.10.2. The port at 10.10.10.1 already appears. The number that follows 10.10.10.1 is the iSCSI port, 3260.

7 Add the second port.

Click **Add**.

The Add Send Targets Server dialog box appears.



Enter the IP address of the second port of the iSCSI array, then click **OK**.

The second iSCSI array port appears in the Dynamic Discovery dialog box.

8 Close the iSCSI Initiator Properties dialog box.

Click **Close**.

9 Be sure the LUNs are visible in VirtualCenter.

Choose the server in VirtualCenter and click the **Configuration** tab. Click the **Storage adaptors** link. Then click **Rescan**.

The LUNs available on the storage array for the newly configured iSCSI adaptor are listed.

The example in this paper uses dynamic discovery. You may use static discovery, if you prefer. For details, see the VMware Infrastructure 3 documentation.

Configuring a Software iSCSI Initiator

Before configuring a software iSCSI initiator, use the Virtual Infrastructure Client to configure the following key settings on your ESX Server host. See your VMware Infrastructure 3 documentation for detailed procedures.

1 Enable VMotion and IP Storage licenses.

2 Configure a VMotion and IP Storage connection in the networking configuration.

3 Add a service console port to the same virtual switch used in step 2.

Make sure both the VMotion and IP Storage network and the service console port connection have appropriate IP addresses and are routed properly to the array.

If you do not take these steps, when you try to add a software iSCSI initiator, you might see an error message that says, “Cannot configure software iSCSI. VMotion and IP Storage has not been enabled.”

IP Storage refers to any form of storage that uses IP network communication as its foundation. For ESX Server, the term refers to both iSCSI and NAS. Because both of these storage types are network-based, both types can use the same port group.

The software initiator works with the ESX Server 3 networking stack, implemented in the VMkernel. The software initiator works with a daemon that runs in the service console. The iSCSI daemon initiates the session, then handles logging in and authentication. After the connection is established, I/O between the storage device and virtual machines on the ESX Server host is handled by the VMkernel. This means the service console and VMkernel Ethernet adapters both need to communicate with the iSCSI storage. In addition, the iSCSI software initiator network and a vswif interface from the service console network must be on the same subnet. The service console can have more than one network port.

As an alternative, you can use routing to give both the service console and the iSCSI VMkernel network interface access to the storage. However, the method described above is an easier option to set up through Virtual Center.

TCP/IP Stack for Software iSCSI

The VMware VMkernel IP networking stack has been extended to handle the following functions:

- iSCSI as a virtual machine datastore (new in ESX Server 3)
- NFS as a virtual machine datastore (new in ESX Server 3)
- NFS for the direct mounting of ISO files, which are presented as CD-ROMs to virtual machines
- Migration with VMotion

The IP address that you assign to the service console during installation must be different from the IP address that you assign to VMkernel's IP stack from the **Configuration** > **Networking** tab of the Virtual Infrastructure Client. The NFS and iSCSI functions must be configured together. They always share the same IP address, gateway, netmask, and other parameters. They are connected to the same virtual switch and, therefore, to the same physical Ethernet adapter. Before configuring software iSCSI for the ESX Server host, you need to open a firewall port. You do so by enabling the iSCSI software client service.

Take the following steps to set up VMotion and IP storage:

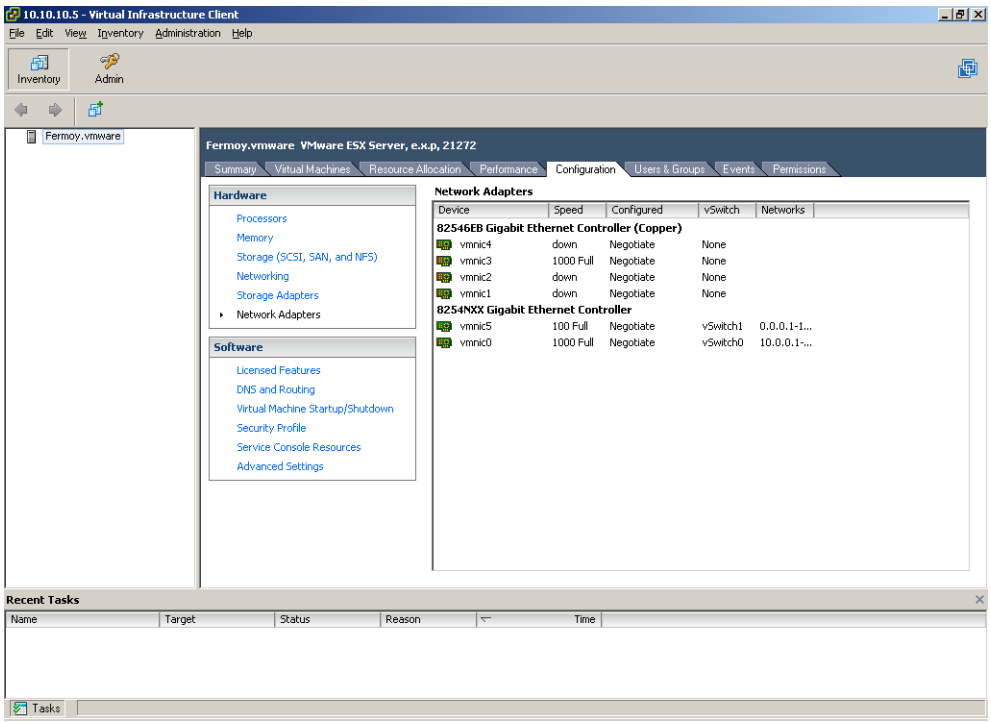
1 Start the VI Client and choose the ESX Server host you want to configure.

2 Click the **Configuration** tab.

3 View the currently available network adapters.

In the **Hardware** list, click **Network Adapters**.

The adapters appear in the list in the right pane. The list provides status information including whether each adapter is up or down.

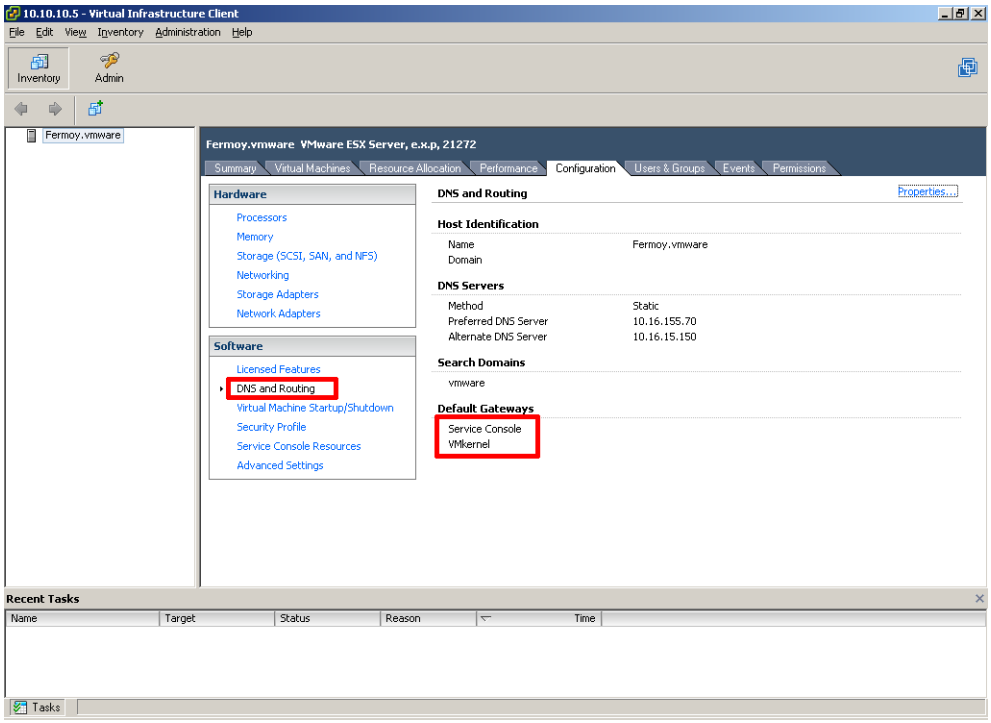


4 Check to see whether VMkernel networking is configured.

In the **Software** list, click **DNS and Routing**.

If the VMkernel has no IP address listed under **Default Gateway**, VMkernel networking is not yet configured. Continue to the next step to configure VMkernel networking.

If the VMkernel has an IP address listed under **Default Gateway**, VMkernel networking is already configured. Stop. You do not need to perform any more steps in this procedure.



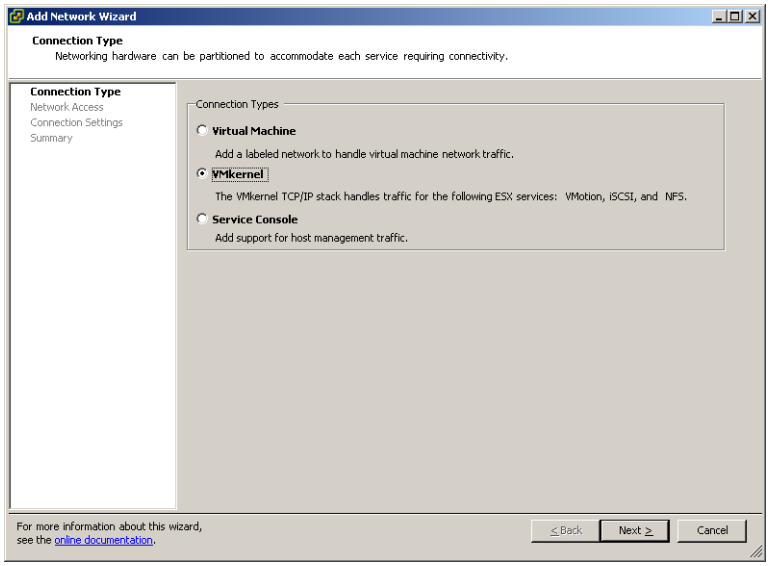
5 Go to the Networking view and add a networking link.

In the **Hardware** list, click **Networking**, then click the **Add Networking** link.

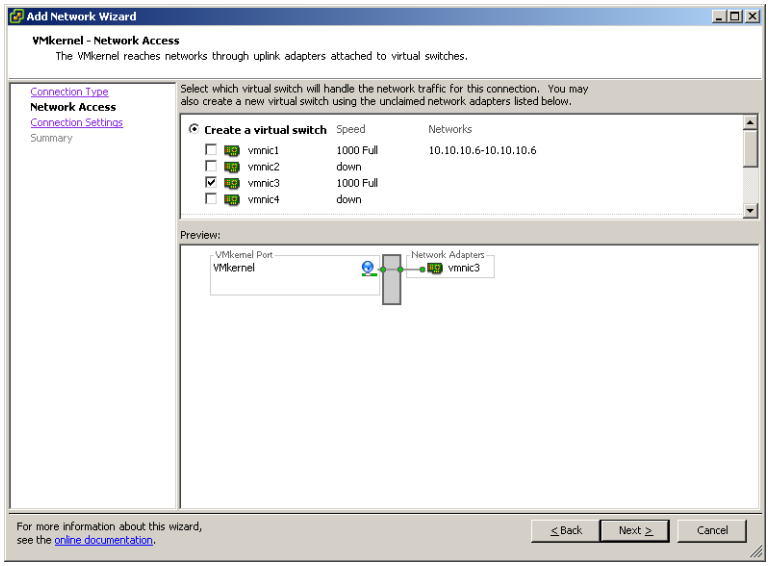
The Add Networking Wizard appears.

6 In the **Connection Type** panel, select **VMkernel**, then click **Next**.

Selecting VMKernel allows you to connect the VMkernel, which runs services for VMotion and IP storage (NFS or iSCSI), to the physical network.



The **Network Access** panel appears.



7 Set up the virtual switch for this connection to use

Select **Create virtual switch**. Use the check boxes to select the physical Ethernet adapters you want the virtual switch to use.

The Preview pane shows your choices.

In this example, vmnic3 is the physical Ethernet adapter that connects to the iSCSI array.

Click **Next**. The **Connection Settings** panel appears.

8 Set the port group properties.

Under **Port Group Properties**, choose or enter a network label and a VLAN ID.

Network Label is a name that identifies the port group you create. This is the label you specify when configuring a virtual adapter to be attached to this port group. You need this label when configuring either virtual machines or VMkernel services, such as VMotion and IP storage.

VLAN ID identifies the VLAN for the port group's network traffic to use.

NOTE Do not select **Use this port group for VMotion**. That selection is used to enable a port group to advertise itself to another ESX Server host as the network connection to which VMotion traffic should be sent. This property can be enabled for only one VMotion and IP Storage port group. If this property is not enabled for any port group, migration with VMotion to this host is not possible. Because this connection is for iSCSI only, it should not be set for VMotion use.

---

9 Set a default gateway for the port you just created.

Under **IP Settings**, click the **Edit** button, then set the VMkernel default gateway for VMkernel services, such as VMotion, NAS, and iSCSI.

NOTE In VMware Infrastructure 3, you must use a valid IP address to configure the VMkernel IP stack, not a dummy address.

Click **OK** to save your settings and return to the wizard, then click **Next**.

The DNS Configuration dialog box appears.

10 Click the **DNS Configuration** tab.

The name of the host appears in the **Name** field by default. The preferred DNS server addresses and the domain also appear. In most cases, the defaults are correct. If they are not, make any necessary changes here.

11 Click the **Routing** tab, and specify gateway information if needed.

The service console and the VMkernel are often not connected to the same network. As a result, each needs its own gateway information. You must specify a gateway if you need connectivity to machines not on the same IP subnet as the service console or VMkernel.

CAUTION If you misconfigure the gateway, the VI Client may lose its connection to the host. If this connection is lost, you must reconfigure the host from the command line at the service console. The **Static IP** setting is the default.

12 Click **OK** to save your changes and close the DNS Configuration dialog box.

13 Click **Next**.

14 Review your changes on the Ready to Complete pane

Click **Back** if you need to make any changes.

Click **Finish** if the configuration is correct.

Configuring an iSCSI Software-Initiated Storage Adapter

Take the following steps to configure an iSCSI software-initiated adapter:

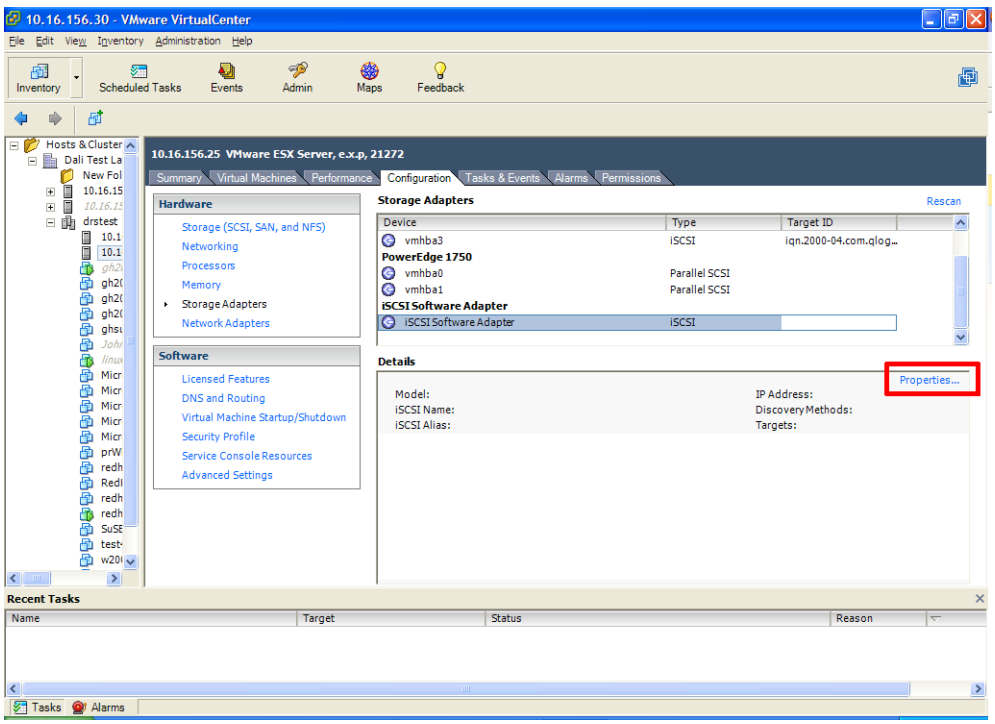
1 Start the VI Client and choose a server from the inventory panel.

2 Click the **Configuration** tab, then click **Storage Adapters**.

The list of available storage adapters appears.

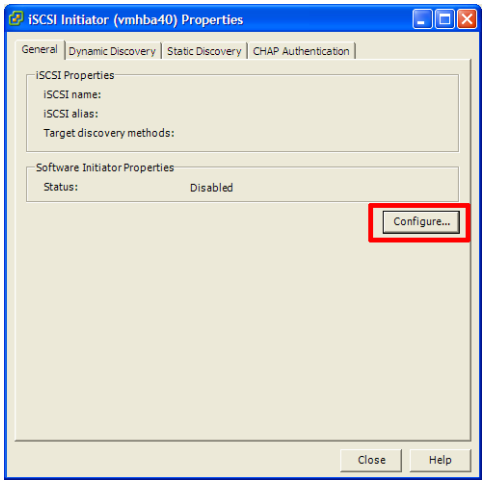
3 Choose one of the iSCSI software adapters.

The **Details** list shows the configuration for the adapter, including the model, IP address, iSCSI name, discovery methods, iSCSI alias, and any discovered targets.



4 Click the **Properties** link in the **Details** pane.

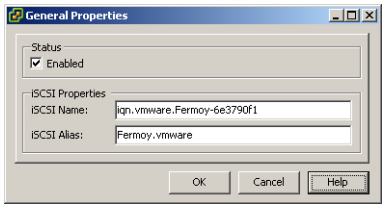
The iSCSI Initiator Properties dialog box appears.



5 Enable the adapter and assign the adapter an alias.

Click **Configure**.

The General Properties dialog box appears, showing the network status and default values for the adapter's iSCSI name and iSCSI alias.



If the software initiator is configured, **Enabled** is selected in the **Status** section of the dialog box.

You can change the default iSCSI name. If you do, make sure you enter a properly formatted name. If the name is not properly formatted, some storage devices may not recognize it. The default name provided by the adapter is properly formatted. However, some array vendors, such as NetApp, look for a name in a different format.

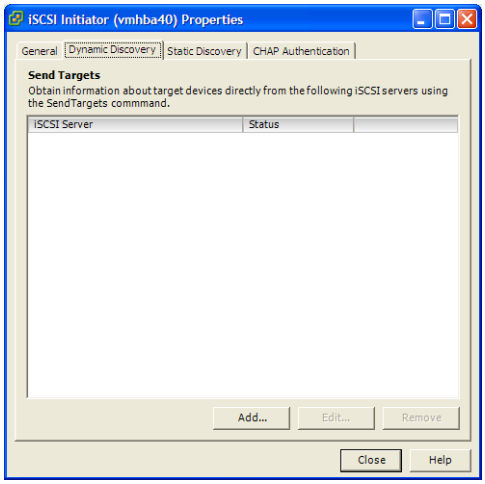
You can change the default iSCSI alias. The iSCSI alias is a friendly name used to identify the iSCSI adapter.

NOTE If you need to change the iSCSI alias later, you can do so using this dialog box. If you change the alias after you first enable the software iSCSI initiator, you must reboot the ESX Server host for the name change to take effect.

---

6 Click **OK** to save your changes.

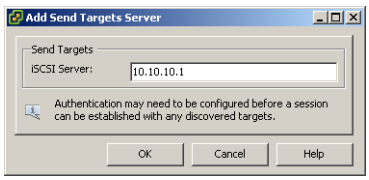
7 To add, edit, or remove dynamic send target discovery addresses, click the **Dynamic Discovery** tab.



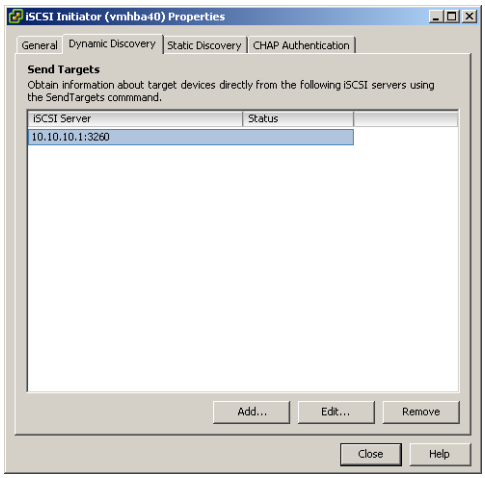
NOTE HBAs attempt to initiate sessions with the list of targets in the dynamic discovery list. An entry in this list does not indicate an active session.

8 To change or delete an existing address, choose the address, then click **Edit** or **Remove**.

9 To add a new send target address, click **Add**. Enter the IP address of the iSCSI server and click **OK**.



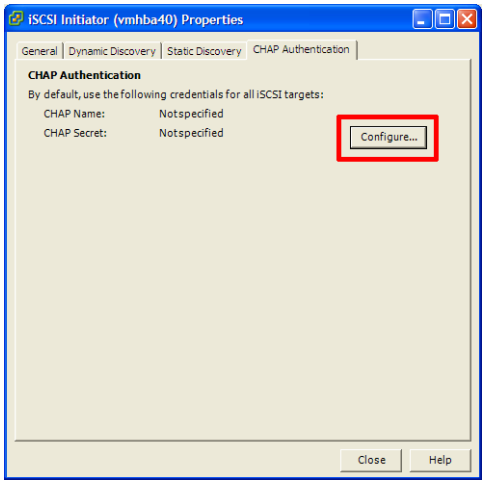
The iSCSI send targets appear in the Dynamic Discovery tab of the iSCSI Initiator Properties dialog box. The number 3260 that follows the IP address in the screen shot below is the iSCSI port number.



NOTE Do not make any changes on the **Static Discovery** tab. Static Discovery is not supported for software iSCSI in VMware Infrastructure 3.

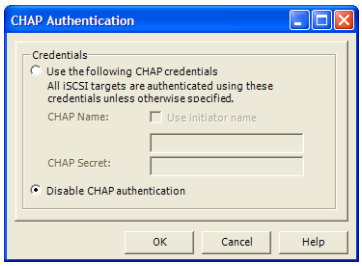
10 To view current CHAP parameters, click the **CHAP Authentication** tab.

The dialog box shows whether CHAP is enabled or disabled.



11 To change the CHAP settings, click **Configure**.

The CHAP Authentication dialog box appears.



To enable CHAP, select **Use the following CHAP credentials**, then enter a CHAP name and CHAP secret.

NOTE If you change this setting, all new targets that attempt to authenticate with the initiator must use the specified CHAP secret to authenticate with the HBA. Any existing sessions are not affected.

To disable CHAP, select **Disable CHAP authentication**.

Click **OK** to save your changes.

NOTE If you disable CHAP, all sessions that required CHAP end immediately when you click OK.

Appendix A - iSCSI Command Line Options and Use

Although VirtualCenter is the preferred method for configuring the iSCSI software and hardware initiators, you can also use the `vmkiscsi-tool` command at the command line. This appendix provides details on some of the options available with this command and provides use case examples for some basic management functions.

```
Usage: vmkiscsi-tool <command> <subcommand> adapterName
Commands:
  -h --help
               -D --discovery
               -S --static: Static Discovery Targets
               -A --Authentication
               -T --Target
               -L --Lun
               -P --Phba
               -N --Network: network properties
               -p --Pnp: Physical Network Portal properties
               -t --ipv4AddrType
               -i --ipv4Address
               -d --dnsserver
               -g --gateway
               -s --subnetmask
               -I --iSCSIname
               -k --Alias
               -e --ethernet: Link Status
               -c --ipconfig: enable/disable DHCP
               -X --Reset
Subcommands:
               -l --list
               -r --remove
               -a --add
               -m --authMethod : specify method for add/remove
               -f --flag: set a discovery or authentication flag


Combine -l with an option to display the current information.

Properties
# vmkiscsi-tool -P -l vmhba3

=========PHBA Properties for Adapter vmhba3=========
VENDOR                      : QLogic
MODEL                       : QLA4010
DESCRIPTION                 : QLogic qla4010 card with a copper interface.
SERIAL NUMBER               : FS20542B11341

=========Node Properties for Adapter vmhba3=========
NODE NAME VALID             : 1
NODE NAME                   : iqn.2000-04.com.qlogic:qla4010.fs20542b11341
NODE ALIAS VALID            : 1
NODE ALIAS                  : iscsi_hba_1
NODE NAME AND ALIAS SETTABLE: 1

Discovery
# vmkiscsi-tool -D -l vmhba3

=========Discovery Properties for Adapter vmhba3=========
iSnsDiscoverySettable    : 0
iSnsDiscoveryEnabled     : 0
staticDiscoverySettable  : 0
staticDiscoveryEnabled   : 1
sendTargetsDiscoverySettable  : 0
sendTargetsDiscoveryEnabled   : 1
slpDiscoverySettable  : 0
Discovery Status: Done.
DISCOVERY ADDRESS          : 0.0.0.0:3260
STATIC DISCOVERY TARGET
```

```
   NAME     : iqn.1992-04.com.emc:ax.ck300060400099.b0
   ADDRESS  : 10.10.10.2:3260
Network
# vmkiscsi-tool -N -l vmhba3
IP CONFIG SETTABLE                    : 1
DHCP CONFIGURATION ENABLED            : 0
SUBNET MASK SETTABLE                  : 1
SUBNET MASK VALID                     : 1
SUBNET MASK                           : 255.255.255.0
DEFAULT GATEWAY SETTABLE              : 1
DEFAULT GATEWAY VALID                 : 1
DEFAULT GATEWAY                       : 10.10.10.254
Primary DNS SETTABLE                  : 1
Primary DNS VALID                     : 1
Primary DNS Address                   : 0.0.0.0
Alternate DNS SETTABLE                : 0
Alternate DNS VALID                   : 0
Alternate DNS Address                 : 0.0.0.0

iSCSI IQN
# vmkiscsi-tool -I -l vmhba3
iSCSI Node Name: iqn.2000-04.com.qlogic:qla4010.fs20542b11341

Ethernet
# vmkiscsi-tool -e -l vmhba3
Ethernet Link Status: Up

Targets & LUNs
# vmkiscsi-tool -T -l vmhba3

-------------------------------------------
NAME                            : iqn.1992-04.com.emc:ax.ck300060400099.b0
ALIAS                           : 0099.b0
DISCOVERY METHOD FLAGS          : 8
SEND TARGETS DISCOVERY SETTABLE : 0
SEND TARGETS DISCOVERY ENABLED  : 0
Portal 0                        : 10.10.10.2:3260

-------------------------------------------
# vmkiscsi-tool -L -l vmhba3

Target iqn.1992-04.com.emc:ax.ck300060400099.b0:
-------------------------------------------
OS DEVICE NAME    : vmhba3:0:0
BUS NUMBER        : 0
TARGET ID         : 0
LUN ID            : 0
Let's see what the HW iSCSI device looks like from the vmkiscsi-tool command on the ESX:
# vmkiscsi-tool -l -D vmhba2
=========Discovery Properties for Adapter vmhba2=========
iSnsDiscoverySettable    : 0
iSnsDiscoveryEnabled     : 0
staticDiscoverySettable  : 0
staticDiscoveryEnabled   : 1
sendTargetsDiscoverySettable  : 0
sendTargetsDiscoveryEnabled   : 1
slpDiscoverySettable  : 0
Discovery Status: Done.
DISCOVERY ADDRESS          : 0.0.0.0:3260
DISCOVERY ADDRESS          : 10.10.10.1:3260
DISCOVERY ADDRESS          : 10.10.10.2:3260
STATIC DISCOVERY TARGET
   NAME     : iqn.1992-04.com.emc:ax.ck300060400099.a0
   ADDRESS  : 10.10.10.1:3260
STATIC DISCOVERY TARGET
   NAME     : iqn.1992-04.com.emc:ax.ck300060400099.b0
   ADDRESS  : 10.10.10.2:3260
```

By listing the discovered targets on vmhba2, we see both iSCSI array ports are discovered. Note the iqn values. The first ends in a0 & the second ends in b0. These represent the Clariion ports.

vmkiscsi-tool outputs from a software iSCSI initiator

```
# vmkiscsi-tool -P -l vmhba40
=========PHBA Properties for Adapter vmhba40=========
VENDOR                      : VMware
MODEL                       : VMware-Isoft
DESCRIPTION                 : VMware Software Initiator
SERIAL NUMBER               :
=========Node Properties for Adapter vmhba40=========
NODE NAME VALID             : 1
NODE NAME                   : iqn.vmware.Fermoy-6e3790f1
NODE ALIAS VALID            : 1
NODE ALIAS                  : Fermoy.vmware
NODE NAME AND ALIAS SETTABLE: 1

# vmkiscsi-tool -D -l vmhba40
=========Discovery Properties for Adapter vmhba40=========
iSnsDiscoverySettable    : 0
iSnsDiscoveryEnabled     : 0
staticDiscoverySettable  : 0
staticDiscoveryEnabled   : 0
sendTargetsDiscoverySettable  : 0
sendTargetsDiscoveryEnabled   : 1
slpDiscoverySettable  : 0
Discovery Status: Done.
DISCOVERY ADDRESS          : 10.10.10.1:3260
DISCOVERY ADDRESS          : 10.10.10.2:3260

Static Discovery not supported for this adapter

# vmkiscsi-tool -I -l vmhba40
iSCSI Node Name: iqn.vmware.Fermoy-6e3790f1

# vmkiscsi-tool -k -l vmhba40
iSCSI Node Alias: Fermoy.vmware

# vmkiscsi-tool -A -l vmhba40
Supported Authentication Methods for Adapter vmhba40:
 AUTHMETHOD_NONE
 AUTHMETHOD_CHAP

# vmkiscsi-tool -T -l vmhba40
-------------------------------------------
NAME                            : iqn.1992-04.com.emc:ax.ck300060400099.a0
ALIAS                           : 0099.a0
DISCOVERY METHOD FLAGS          : 0
SEND TARGETS DISCOVERY SETTABLE   : 0
SEND TARGETS DISCOVERY ENABLED    : 0
Portal 0                        : 10.10.10.1:3260

-------------------------------------------
NAME                            : iqn.1992-04.com.emc:ax.ck300060400099.b0
ALIAS                           : 0099.b0
DISCOVERY METHOD FLAGS          : 0
SEND TARGETS DISCOVERY SETTABLE   : 0
SEND TARGETS DISCOVERY ENABLED    : 0
Portal 0                        : 10.10.10.2:3260

-------------------------------------------
```

```
# vmkiscsi-tool -L -l vmhba40
Target iqn.1992-04.com.emc:ax.ck300060400099.a0:
-------------------------------------------
OS DEVICE NAME    : vmhba40:0:0
BUS NUMBER        : 0
TARGET ID         : 0
LUN ID            : 0

-------------------------------------------

Target iqn.1992-04.com.emc:ax.ck300060400099.b0:
-------------------------------------------
OS DEVICE NAME    : vmhba40:1:0
BUS NUMBER        : 0
TARGET ID         : 1
LUN ID            : 0
```


Revision 20071004 Item: DG-021-INF-01-01