Copyright Statement

Chapter 3: Implementing an iSCSI Storage System

Chapter 1 covered some of the important Exchange Server concepts relating to databases and transaction logs as well as discussed factors that may influence your decision in choosing between direct attached storage (DAS) or networked storage such as a storage area network (SAN). Choosing the right platform on which to store Exchange Server databases and transaction logs will help you to support the necessary growth you will require for the future.

Many Exchange administrators view Exchange storage sizing as merely adding the maximum amount of disk space that a server can support. For a number of reasons, this is not the best logic because you may be insufficiently estimating either the necessary storage capacity or the disk I/O capacity. Chapter 2 discussed two important concepts that are crucial when planning your disk system capacity; these include adequately estimating the amount of space the databases and indexes will consume and the disk space necessary to support transaction logs, and finally ensuring that the disk subsystem is capable of supporting the necessary I/O capacity. Chapter 2 also discussed the concept of I/Os per second (IOPS) and the importance of considering the expected disk I/O load the users will put on the disk subsystem.

This chapter will introduce the concepts of an iSCSI-based SAN storage system and discuss how to implement iSCSI from the Windows Server perspective when using the Microsoft iSCSI initiator. Understanding not only the implementation of iSCSI but also some of the concepts surrounding SCSI in general will help you to better understand how this works in your environment. Important topics in this chapter include:

- Understanding the basics of iSCSI
- Installing and configuring the Microsoft iSCSI initiator
- Possible network designs for iSCSI implementations
- Moving Exchange data and logs to iSCSI logical units (LUNs)
- Hardware solutions and compatibility

iSCSI Basics and Terminology

Let’s start with a basic discussion of SCSI, iSCSI, and the appropriate standards. One thing that makes systems administrators reluctant to even consider an implementation of an iSCSI-based SAN is that they are concerned that they are implementing a proprietary system. These fears are unfounded.

Basic SCSI (or Small Computer Systems Interface, but pronounced skuzzy) was developed by Larry Boucher who later founded Adaptec; SCSI was first widely adopted in the desktop world by Apple. Originally, SCSI defined a 50-pin parallel interface for connecting computers, disks, tapes, scanners, optical media, and other devices at a maximum speed of a whopping 10MBps. SCSI has been widely adopted over the years on many platforms. The standard has been expanded so that the current SCSI-3 protocol can be used over other media such as Fibre Channel connections, FireWire connections (using the Serial Bus Protocol), and IP networks.

iSCSI (pronounced eye-scuzzy) is an implementation of the SCSI protocol that transports SCSI communication and data over IP networks; it is an official Internet standard adopted by the Internet Engineering Task Force (IETF) in RFCs 3720 and 3783. Though this standard was ratified in 2003, it was not immediately adopted for widespread use. However, Gigabit Ethernet has become increasingly common and affordable and thus has encouraged acceptance of iSCSI.

The typical iSCSI environment is illustrated in Figure 3.1. In any iSCSI environment, you have targets and initiators. The initiator is the client and the target is the host that is providing the LUN; typically, the target LUN is disk storage, but in practical terms, it could be a tape device, optical disk, or other device on the target system.

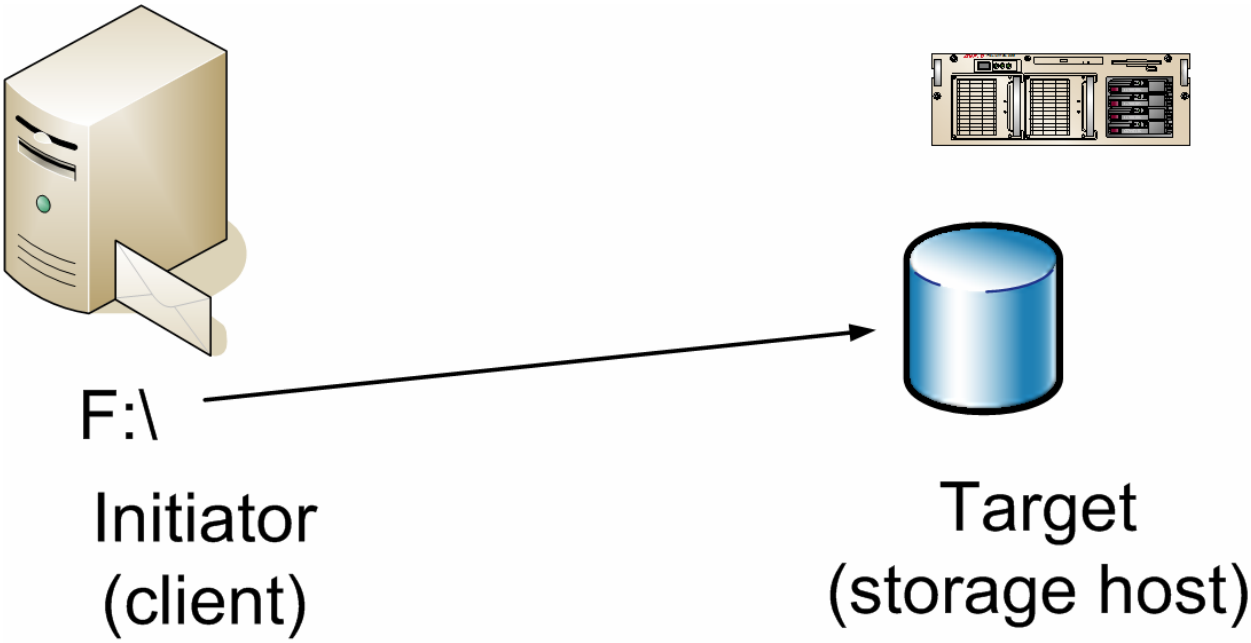

Figure 3.1: Initiator and target illustrated.

In some cases, the initiator is software that is installed on the client operating system (OS) such as the Microsoft iSCSI Software Initiator. However, the initiator may be combination of hardware and software; in this case, the hardware may be a dedicated iSCSI adapter that provides boot and TCP offload processing capabilities as well as the device drivers and software to manage the iSCSI adapter. Hardware-based iSCSI adapters can provide higher levels of performance than a software-only solution because a hardware iSCSI adapter can offload much of the processing related to network communication to the memory and processor on the hardware adapter.

The target system is any type of networked storage or SAN that supports the iSCSI protocol and allows the target's local resources (usually disk volumes) to be assigned to a LUN that can be accessed through the iSCSI protocol. The LUN represents dedicated storage that has been allocated or carved out of the available disk storage on the target system; the LUN will not be in use by more than one initiator at a time. In simple storage configuration, the LUN will be assigned to only a single initiator; however, the exception to this is in the case of clustered nodes. Each node of the cluster must be able to access a LUN on which data may reside and that the cluster must access.

Once connected to a remote LUN, the LUN appears to be locally attached storage on the target to applications such as Exchange Server, SQL Server, or even file sharing.

This initiator/target architecture that uses a combination of well-defined and widely used architectures (SCSI-3 and TCP/IP) allows many organizations that would not have been able to use networked storage otherwise to take advantage of it. Part of this is due to cost and part is due to expertise. Managing a large-scale SAN system requires quite a bit of additional expertise and the Fibre Channel infrastructure for connectivity to the SAN is costly for small and midsized businesses. iSCSI simplifies the connectivity and has allowed storage vendors to provide simpler solutions for not only larger businesses but also small and midsized businesses. iSCSI allows organizations to leverage their existing knowledge (and possibly infrastructure) when deploying the IP network infrastructure necessary.

In addition, iSCSI can provide a higher level of data transport security than Fibre Channel if both the initiator and target systems support the IPSec protocol. Using IPSec, connections between the initiator and the target can be both authenticated and/or encrypted.

Redundancy in connectivity between the iSCSI target and the initiators can be achieved simply and cost effectively with additional networks. Provided both the initiator and the target support multi-path I/O, the initiator can then use multiple networks to connect to the target. This, of course, requires that both the initiator and target systems have multiple network adapters.

Possible iSCSI Designs

Although deploying iSCSI is reasonably simple to do, there are a number of ways that it can be deployed. Let's look at a couple of possible design considerations. The first of these is by far the simplest and is shown in Figure 3.2. In this solution, the iSCSI SAN is connected to the same network switch as the rest of the network components. This switch could be a 10MB, 100MB, or 1GB Ethernet switch.

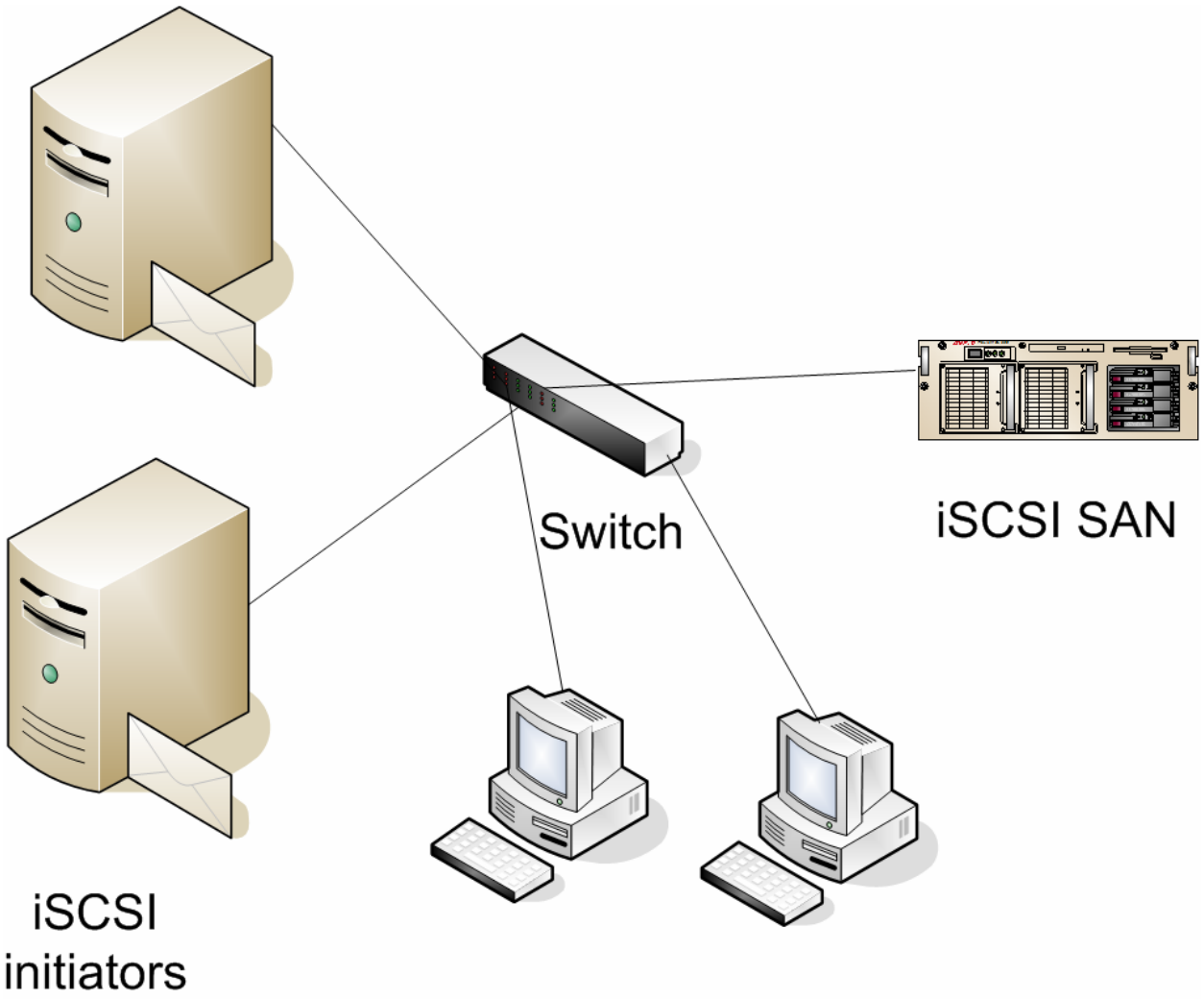

Figure 3.2: Simple iSCSI SAN solution.

This design provides an example of how simple an iSCSI deployment could be, but it is certainly not a best practice. One reason is that storage-related iSCSI traffic will need to share the same network infrastructure as the rest of the computers on the network. Further, this design does not provide multiple paths between the initiators and the iSCSI SAN. This solution might work fine in a small environment or for a lab. However, sharing a network with other traffic will introduce latency into your environment and negatively impact iSCSI performance. Many data networks are also only 10/100Mbps networks; 10/100Mbps networks will work for iSCSI but performance will be quite poor.

Figure 3.3 shows a more practical approach to planning a network for iSCSI. In this example, a dedicated 1GB Ethernet switch is provided. Each node on the network that requires iSCSI LUNs is connected to this dedicated network as well as to the production LAN.

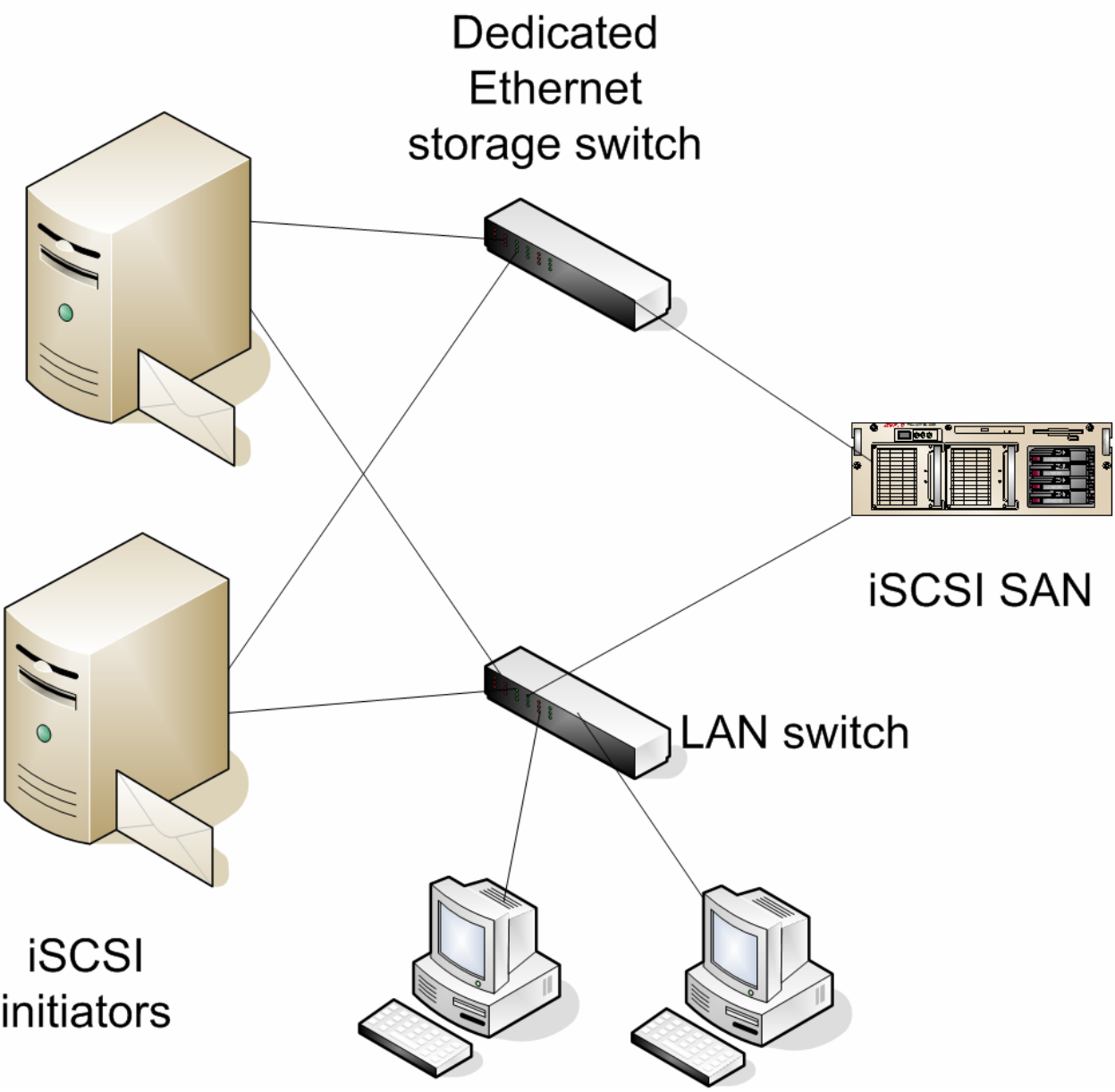

Figure 3.3: Providing a dedicated network for iSCSI storage.

This design can be scaled to allow multi-path I/O by adding dedicated Ethernet switches for the storage network or by allowing the production LAN to be used as a backup path in case the dedicated Ethernet network is not available.

In most cases, each iSCSI client system has a local disk that is used by the OS for booting and for the page file as well as locally installed applications. In some situations, though, it is possible to boot from a SAN LUN. Although this is not very common, sometimes companies want to know if this is possible. Implementing SAN boot introduces additional complexity into your environment, such as having to configure the LUNs and the adapters for SAN boot, but it can allow for faster recovery from hardware failures. If this is a requirement, the iSCSI client must have a bootable iSCSI network interface card or use a PXE client such as emBoot. Bootable network interface cards for iSCSI are available from vendors such as Broadcom and Intel.

In environments in which the iSCSI client will be running applications that are I/O intensive (such as Exchange Server systems with more than 500 mailboxes), iSCSI performance can be improved by using adapters that offload the overhead of the iSCSI protocol or TCP on to a host bus adapter (HBA) card for iSCSI. Vendors such as QLogic, Adaptec, and Alacritec make iSCSI HBAs. Prior to making a design decision to use iSCSI HBAs, consult your iSCSI SAN vendor to review your expected and projected I/O loads to determine whether iSCSI HBAs are necessary. Even with moderately heavy iSCSI loads, the performance on the software-based iSCSI initiator is very good.

Supporting the Windows iSCSI Initiator

Many of the steps required to install and configure an initiator are going to depend on the type of initiator client you are using (hardware or software) as well as the iSCSI target. The examples in this chapter will cover the steps necessary to use the Microsoft iSCSI Software Initiator.

In the examples that require an iSCSI target, I am using a virtual machine running OpenFiler v1.1. OpenFiler is very easy to use and a great way to learn both NAS and SAN-based technologies; LUNs can be created for not only iSCSI initiators but also NFS, SMB/CIFS, FTP, and HTTP-based file access. You can download this virtual machine from http://www.openfiler.com.

To better illustrate the configuration that I am using for the examples in this chapter, Figure 3.4 shows the initiator and target systems. The client or initiator system is an Exchange Server named SFOEX01 whose IP address is 192.168.15.254. The target system is an iSCSI SAN (in this case, running OpenFiler) whose IP address is 192.168.15.128.

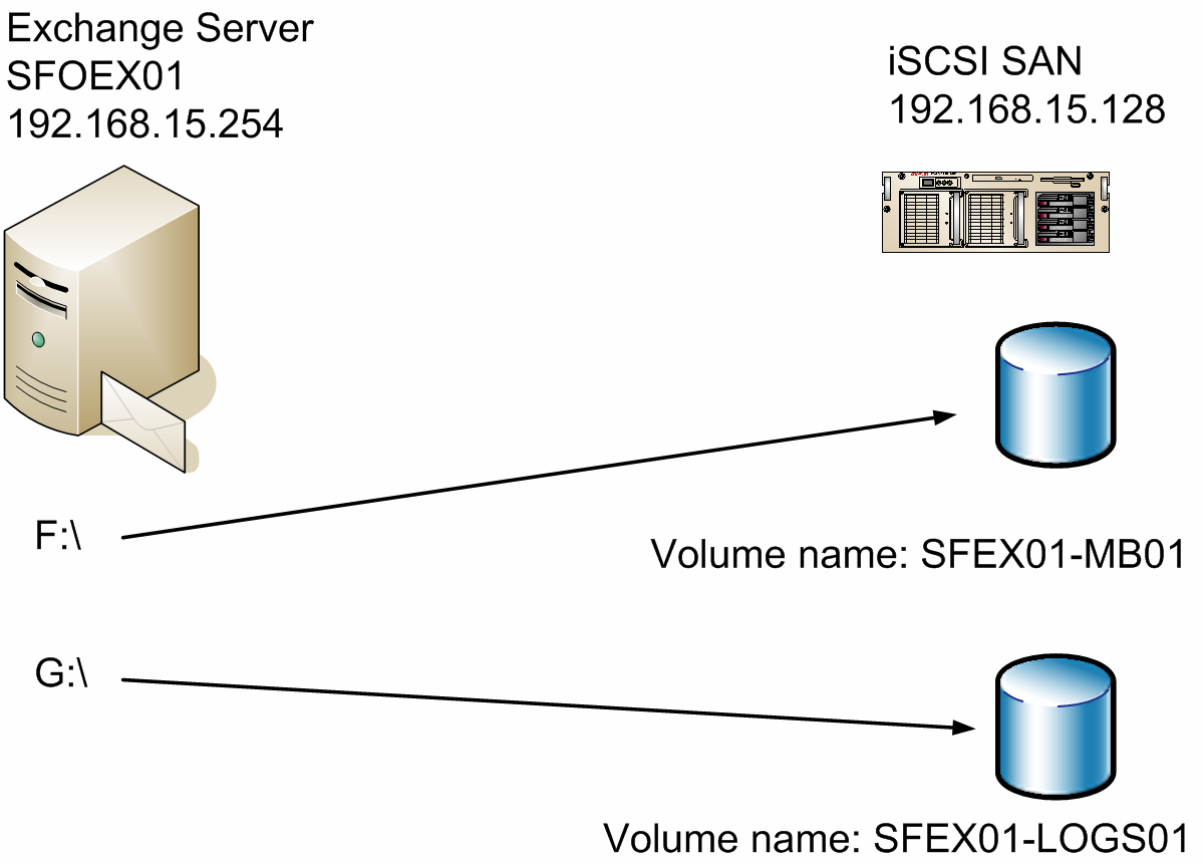

Figure 3.4: Example of an iSCSI system.

Two volumes have been created on the target system and have been configured to be used with iSCSI clients. These volumes are for the Exchange database files and the Exchange transaction log files.

Despite sounding like a complex process, getting Windows 2003 to use iSCSI LUNs is a pretty simple process. The following list summarizes the steps taken in this example:

1. Create volumes on the iSCSI SAN and designate them as iSCSI volumes.
2. Define iSCSI volume security (user name, password, and IP restrictions) for the iSCSI volumes on the iSCSI SAN.
3. Install the iSCSI initiator client software on the Windows server.
4. Connect the iSCSI initiator to the appropriate target LUNs on the iSCSI SAN.
5. Partition and format the new volumes on the Windows server.
6. Move the Exchange databases and transaction logs to the iSCSI LUNs.

Installing the iSNS

Before you install the first iSCSI initiator, you may want to install the Microsoft iSNS Server. iSNS is Internet Storage Name Service, which allows for management and control of iSCSI initiator clients that support a minimum of version 22 of the IETF iSNS draft standard. The iSNS Server installs as a Windows service and allows iSNS clients such as the Microsoft iSCSI initiator to register with the iSNS server and to locate other iSNS clients on the network. The software includes the iSNS service, a graphical user interface (GUI) that allows administration of most common iSNS server management tasks, a command-line interface for managing the iSNS server, and a Windows Management Instrumentation (WMI) interface for managing the iSNS server.

The iSNS server is not necessary and is designed merely to make it easier to discover and manage iSNS clients. In the examples I am going through, I will not use the iSNS client component of the Microsoft iSCSI initiator. If you have more than two or three initiators and targets, the iSNS Server can simplify the discovery and management process.

> 🕮 See http://www.microsoft.com/downloads/details.aspx?familyid=0DBC4AF5-9410-4080-A545-F90B45650E20&displaylang=en or search Microsoft's Web site for iSNS for more information.

Installing the iSCSI Initiator

One of the first things that you have to do in order to use an iSCSI SAN is to install the iSCSI initiator software on your client computers. In the case of client computers, I mean the Windows servers that will be using iSCSI targets; this will almost always be your Windows servers. Unless your SAN or iSCSI network adapter vendor specifically instructs you to do otherwise, always use the latest version of the Microsoft iSCSI Software Initiator. This software can be downloaded from Microsoft's Web site.

> 🕮 For download links and more information about Microsoft's iSCSI Software Initiator, see http://www.microsoft.com/WindowsServer2003/technologies/storage/iscsi/default.mspx.

The Windows iSCSI Software Initiator is easy to install. Follow these steps to install the initiator on to your Windows server:

1. Download the correct build for your Windows Server platform; builds for the x86, x64 (AMD64), and IA64 platforms are available. The installer will have a name such as Initiator-2.04-build3273-amd64fre.exe.
2. Run the initiator client installation on the Windows Server on which you want to install the software. You may have to confirm Run on the Open File - Security Warning dialog box once you run the executable.
3. On the Microsoft iSCSI Initiator welcome screen, click Next.
4. On the Microsoft iSCSI Initiator Installation box (shown in Figure 3.5), select the installation options. I recommend installing all the available options including the Microsoft MPIO Multipathing Support for iSCSI even if you don't plan to use it right away. When you have selected the desired options, click Next.

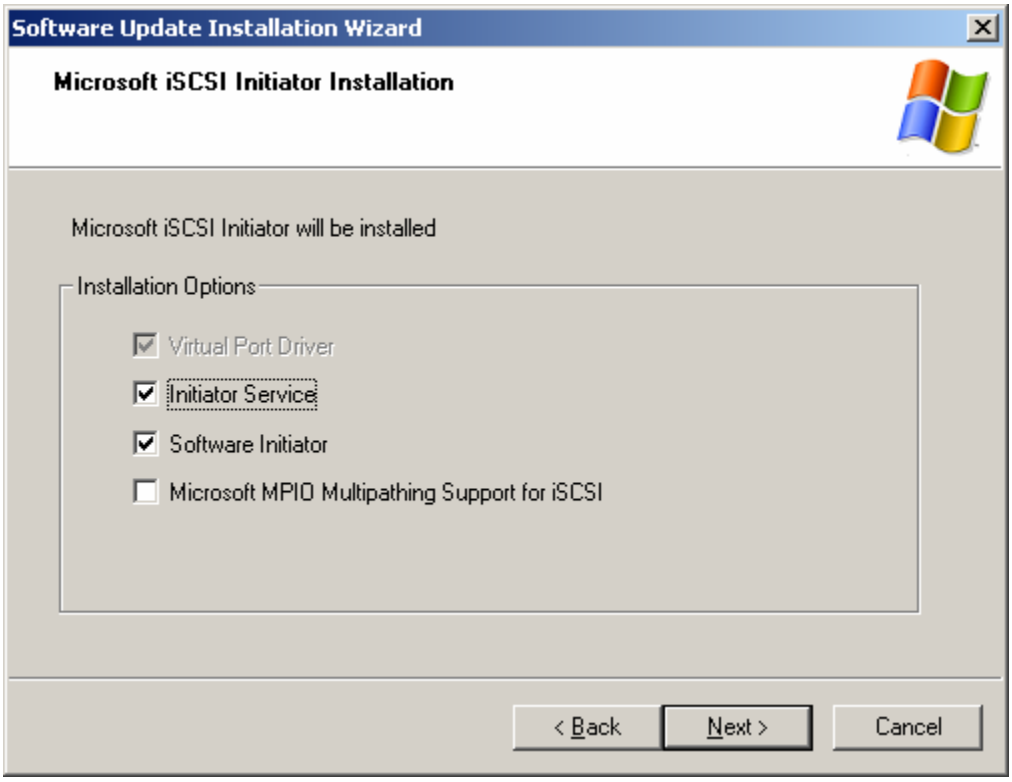

Figure 3.5: Selecting iSCSI Initiator installation options.

5. On the License Agreement screen, review the End-User License Agreement, select the I Agree radio button, and then click Next.
6. The installation may take a minute or two to run all the necessary processes and complete the installation, so be patient. Once the installation has completed, click Finish. Once you click Finish, the system will restart.

One of the most important things to keep in mind is that anytime you install or update the iSCSI initiator software, make absolutely sure that you restart the system immediately after the installation completes. After reboot, you will have a new Control Panel applet available called the iSCSI Initiator (see Figure 3.6). This icon is also placed on the desktop.

Figure 3.6: iSCSI Initiator Control Panel icon.

Configuring the Initiator to Connect to iSCSI LUNs

After installation of the iSCSI Initiator and rebooting the Windows server, you are now to connect the initiator to iSCSI target LUNs. In this example, I have created two volumes on the iSCSI SAN that are part of a volume group named SF-VG01; the volumes are SFEX01-MB01 and SFEX01-LOGS01 (see Figure 3.7).

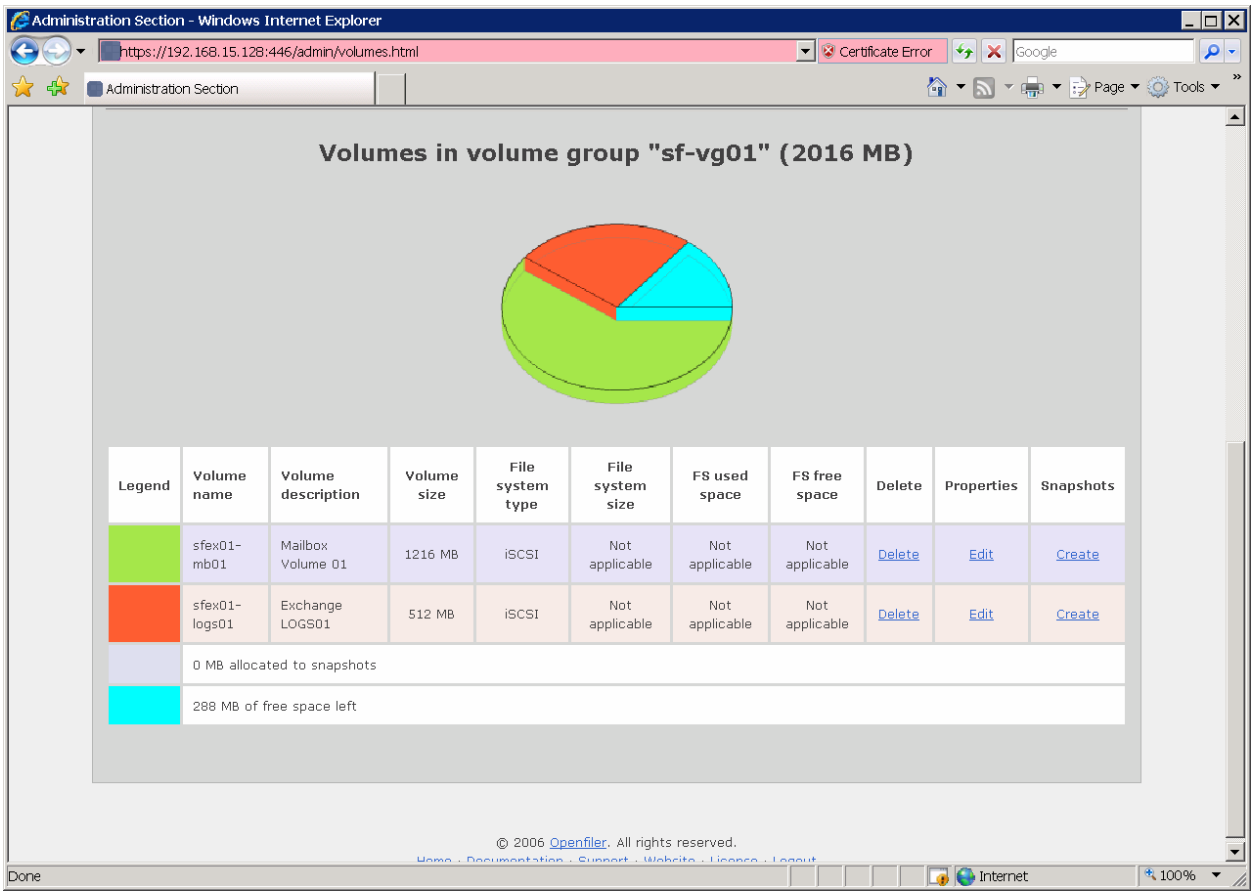

Figure 3.7: Allocating volumes on the Open Filer virtual machine.

On the Open Filer virtual SAN, I need to edit each iSCSI volume that I have created and allow access from a particular IP subnet. In this case, I'm going to allow access to the iSCSI volume from computers on the same subnet (see Figure 3.8).

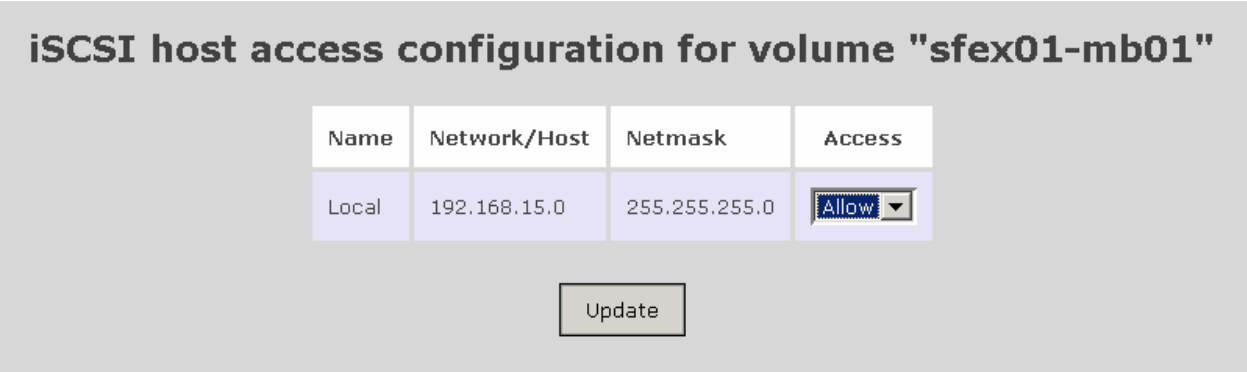

Figure 3.8: Allowing access to the iSCSI volume.

The ability to configure and allow access to a LUN based on IP address will vary from one iSCSI SAN to another, so refer to your iSCSI SAN vendor or documentation to determine whether this step is necessary and how to perform it. Alternatively, I could also configure CHAP authentication to the iSCSI volumes, but that is not necessary in this configuration.

Next, we need to configure the iSCSI initiator. To do so, locate the iSCSI Initiator in Control Panel (or on the Desktop), and run it. Figure 3.9 shows the General property page of the iSCSI Initiator; notice the Initiator Node Name.

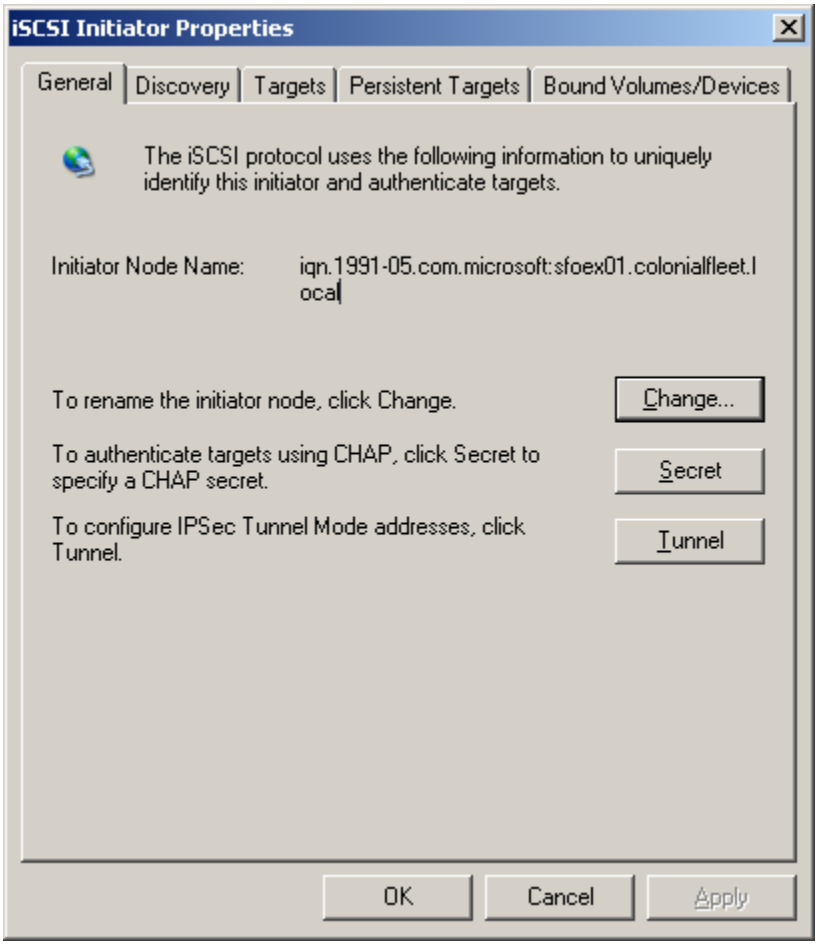

Figure 3.9: General property page of the iSCSI Initiator.

The Initiator Node Name is automatically generated and is intended to be unique on your network. The node name in Figure 3.9 is iqn.1991-05.com.microsoft:sfoex01.colonialfleet.local. This includes the DNS domain name of the Windows server as well as additional information placed there by the iSCSI initiator. iSCSI initiator and target names that start with iqn are called iSCSI qualified names. The format for the first part of an iqn name is iqn.year.month.reversedomain followed by a colon, followed by the DNS domain name of the host or initiator. If the name is part of a target volume, it will include the volume group and the volume name.

> ☞ Though you can change the Initiator Node Name, I recommend keeping it at the default to keep things as simple as possible in your configuration.

The first thing we need to do is to locate the iSCSI targets we are going to be using for this Windows server. Figure 3.10 shows the Discovery property page of the iSCSI Initiator Control Panel application. From the Discovery property page, I specify either the iSNS servers I will use to locate targets and initiators or I can specify the IP addresses of the iSCSI SAN.

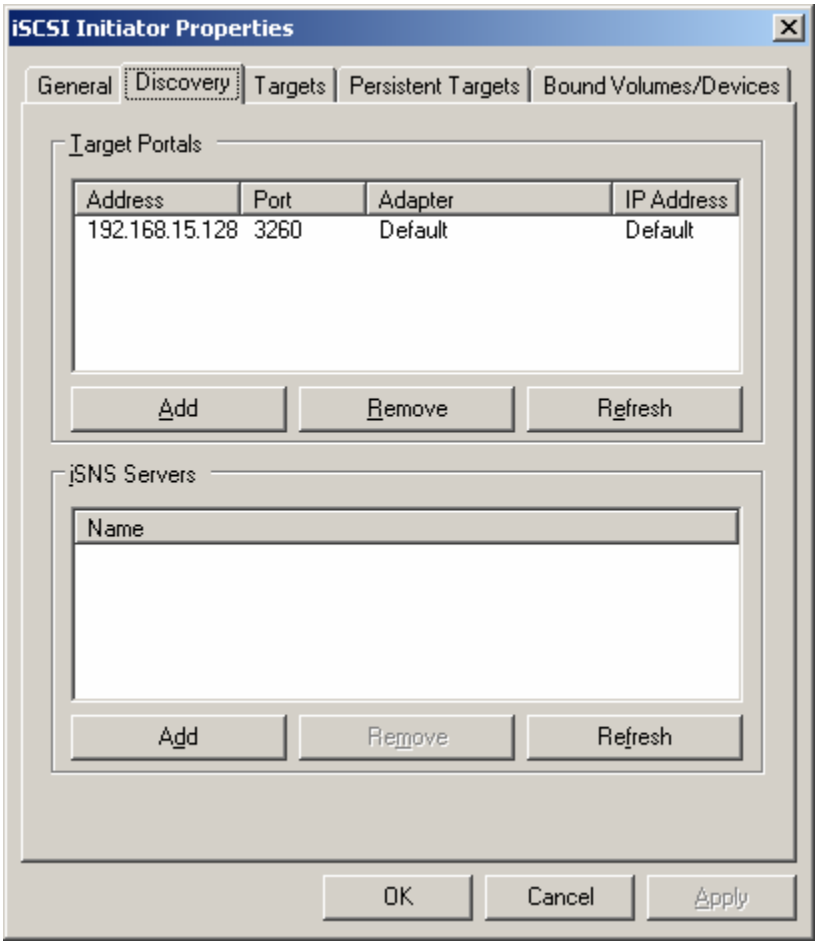

Figure 3.10: Locating the iSCSI targets.

As Figure 3.10 shows, I manually specify the IP address of the iSCSI target I am using for this example. Under the Target Portals section, click Add to see the interface that allows you to add iSCSI SANs. Figure 3.11 shows the Add Target Portal dialog box. All that is actually required is the IP address or DNS name of the iSCSI SAN and the remote TCP port number (3260 is the default).

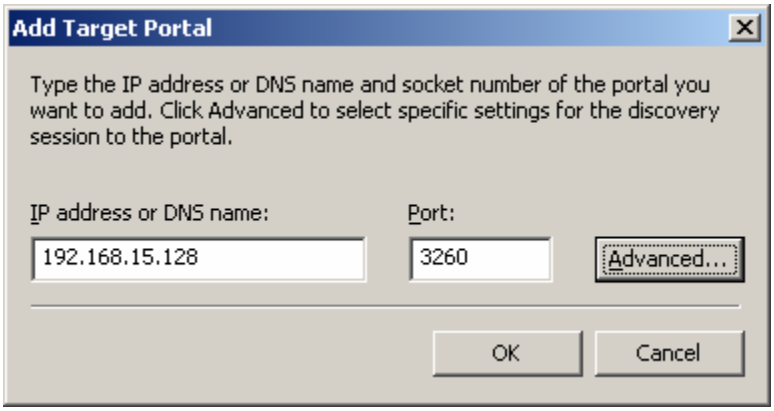

Figure 3.11: The Add Target Portal dialog box.

If you click Advanced, you will see the Advanced Settings property page (see Figure 3.12). For a small environment or a lab setup, you can keep all the default settings found on the Advanced Settings. One time you might need the Advanced Settings property page is when the target system requires CHAP authentication or mutual authentication. CHAP authentication provides very basic security while mutual authentication not only requires the initiator to authenticate with the target but also the target to authenticate with the initiator.

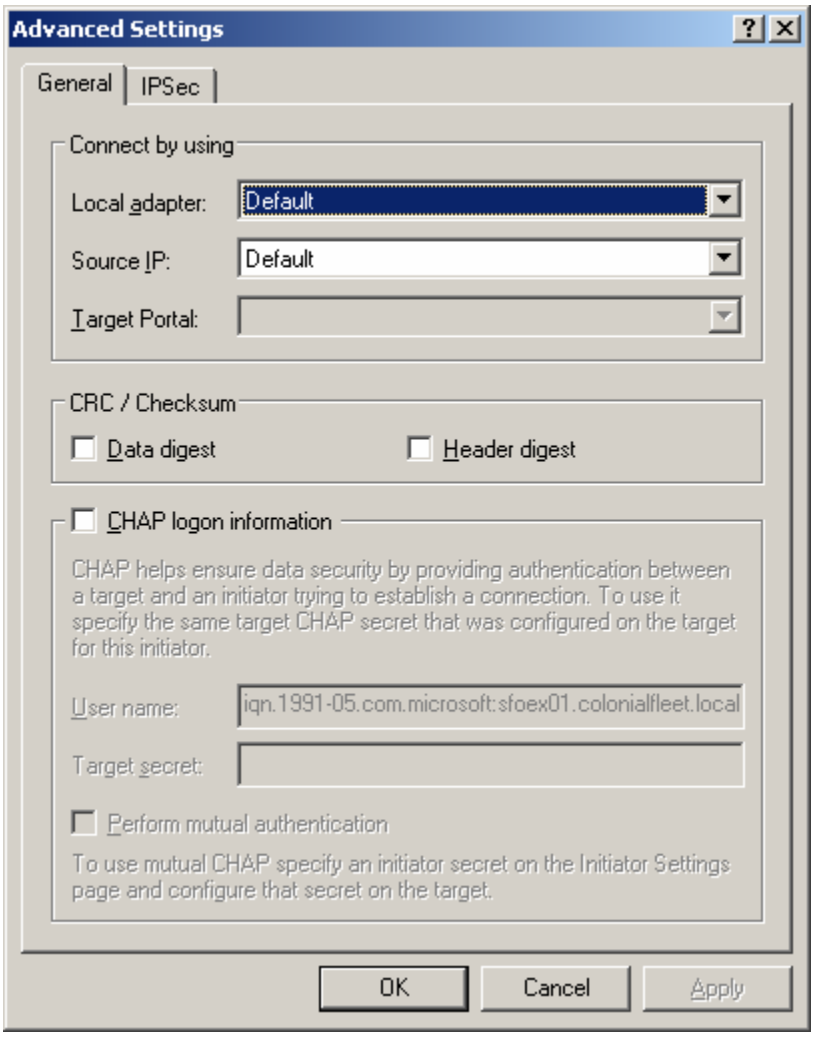

Figure 3.12: Advanced settings when configuring a target portal.

If you are configuring CHAP authentication, the default user name is the local Initiator Node Name. You must also specify a shared secret that must also be configured on the target volume. If the remote iSCSI system must also authenticate the connection, you can specify mutual authentication.

If you are concerned about network-level security (encryption or authentication), enable IPSec between the initiator and the target systems. The Windows iSCSI Initiator supports IPSec, but not all iSCSI SANs do, so check with your vendor.

> ☞ Always follow the iSCSI Initiator procedures provided by your iSCSI SAN vendor. Vendor-provided procedures will always be more accurate than the generic advice I am giving you here or that you will find in Microsoft's documentation.

Once you have finished configuring the target portal information, you can move on to the Targets property page (see Figure 3.13). From here, you can see all the targets that were discovered when you specified target portals on the Discovery page. If the targets for your remote iSCSI SAN do not show up, click Refresh. If they still do not show up, you might have an authentication issue or the IP address of the initiator is not allowed to connect to the target volume.

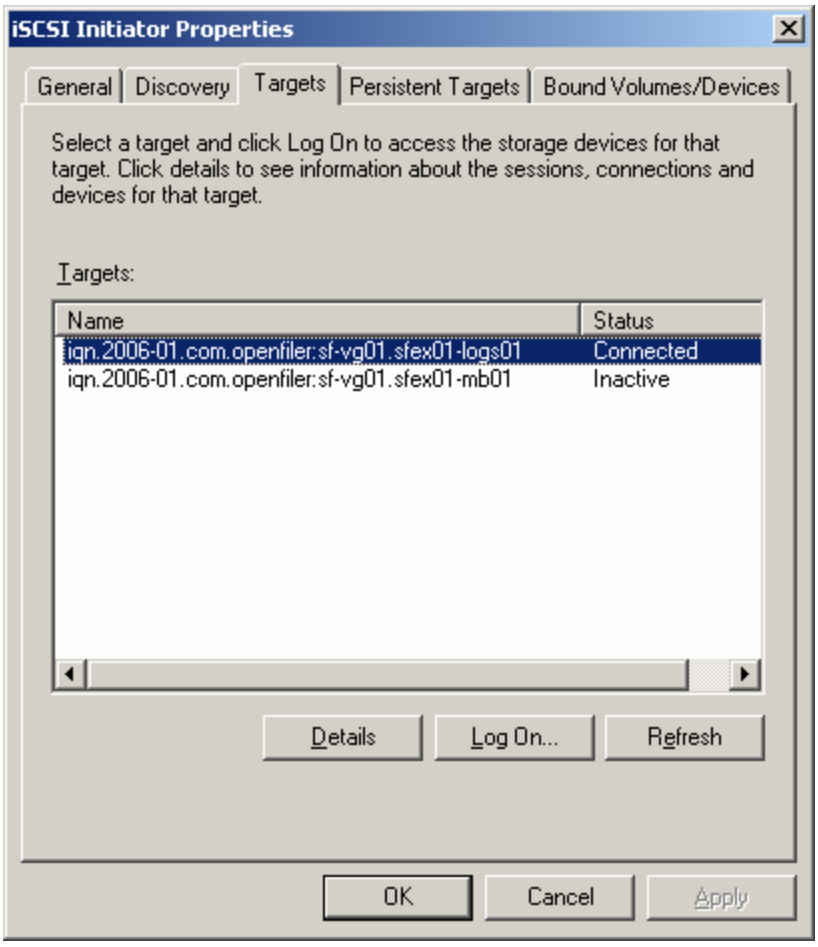

Figure 3.13: Configuring remote iSCSI targets.

By default, the Status of these targets will be Inactive. For each of the targets you want to use, you will need to select each one and click Log On. Notice the target name of this volume is iqn.2006-01.com.openfiler:sf-vg01.sfex01-logs01; this name includes the volume group name (sf-vg01) and the volume name (sfex01-logs01).

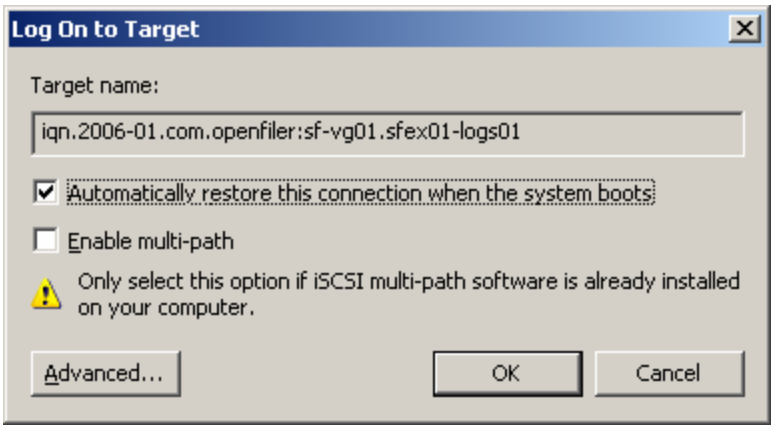

Figure 3.14: The Log On to Target dialog box.

If you are setting up iSCSI volumes for use with applications that require the volume be available when the application starts, such as Exchange or SQL Server, make sure that you select the *Automatically restore this connection when the system boots* check box. Doing so ensures that the target is connected on reboot; this is also called a persistent connection.

> From the iSCSI Initiator Control Panel application, you can remove target volumes. You can also remove the volumes completely by stopping the Microsoft iSCSI Initiator Service. Under no circumstances should you do so while Exchange or SQL databases are active and using iSCSI targets. This will corrupt the data residing on these LUNs just as if you had pulled the plug on a physical disk drive.

After you have connected the targets to this system, it is almost time to partition and format the new volumes that are available to this Windows server. However, before you do so, it is a good idea to get the information necessary to uniquely identify each of the iSCSI volumes. This is not necessary if you are only using one or two target volumes or if they are different sizes, but trust me, this will become valuable on a larger server.

First, take note of the target name, specifically the part that shows the volume name that was created on the iSCSI SAN. In this example, the volume name is SFEX01-MB01. On the Targets property page of the iSCSI Initiator, select each target and click Details. Then select the Devices property page and click Advanced; this shows the Device Details properties. On the Device Details General property page (see Figure 3.15), take note of the SCSI Address information; in this case, Port 2, Bus 0, Target Id 1, LUN 0.

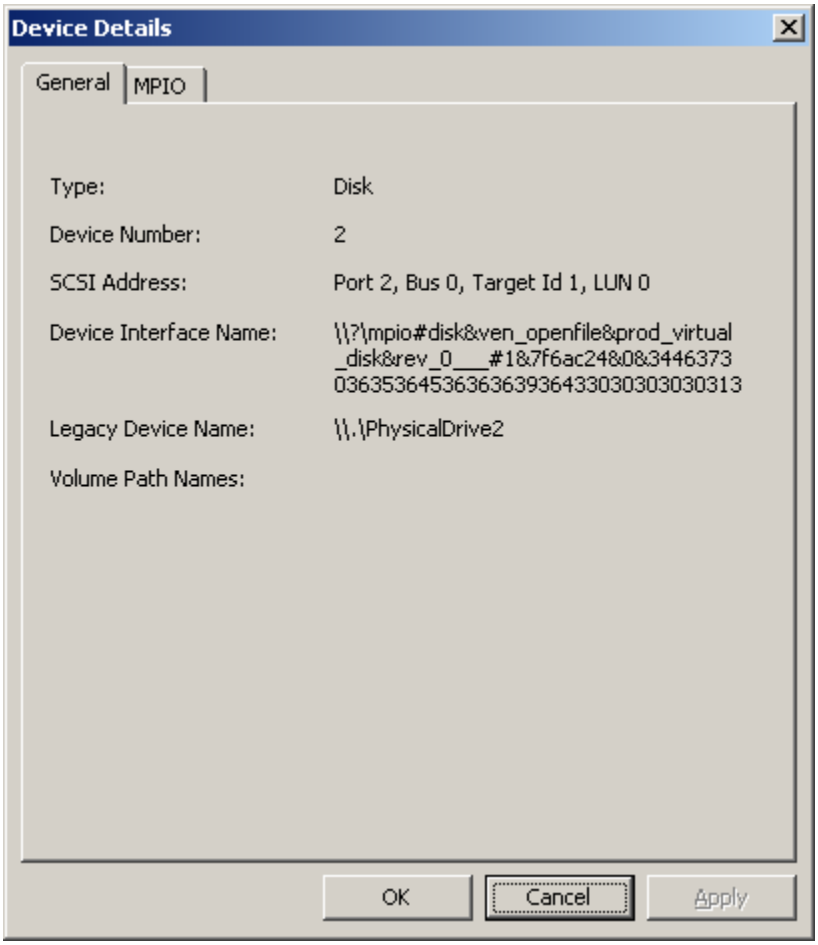

Figure 3.15: Viewing the device details including the SCSI Address.

In the case of this example, I would create a table for this server that documents the SCSI Address and the volume name. Table 3.1 shows an example table for server SFOEX01.

iSCSI SAN Volume Name	Volume Size	Local SCSI Address	Device Id
SFEX01-LOGS01	500MB	Port 2, Bus 0, Target Id 0, LUN 0	1
SFEX01-MB01	1.2GB	Port 2, Bus 0, Target Id 1, LUN 0	2

Table 3.1: SAN volumes and SCSI addresses.

Once you have documented the SCSI Addresses of all your targets, it is time to move on to the Windows Disk Management console (found in Computer Management). From here, we'll partition and format the disks. When you first launch the Disk Management console, Disk Management will recognize there are new disks that do not have a disk type associated.

> ☞ iSCSI-based volumes should always be configured as basic disks. Do not initialize them as dynamic disks.

The Initialize and Convert Disk Wizard will automatically run; you should select Next to continue with the wizard. On the Select Disks To Initialize screen (see Figure 3.16), make sure that all unrecognized disks are selected, then click Next.

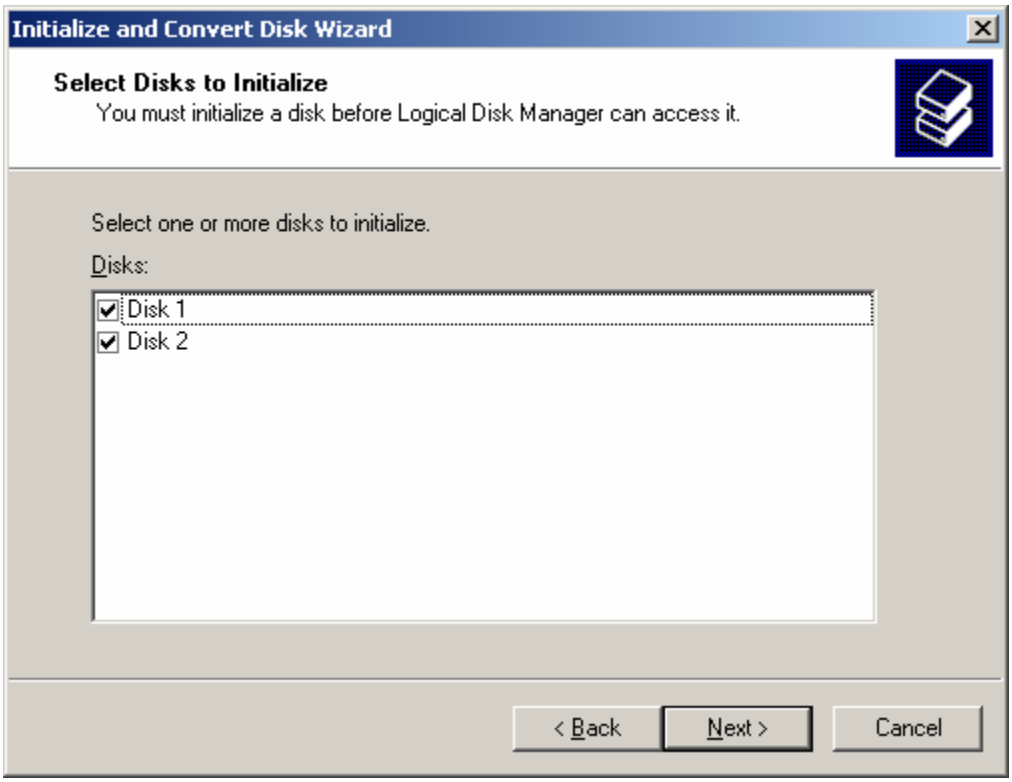

Figure 3.16: Selecting disks to initialize.

Figure 3.17 shows the next screen of the wizard—the Select Disks to Convert screen. By default, the disks will be initialized as basic disks; however, this screen gives you the option of configuring these disks as dynamic disks. Ensure that the new iSCSI disks are *not* selected to be converted, and click Next. SAN disks should always be basic disks; this is true also for any volume that will be part of a Windows cluster.

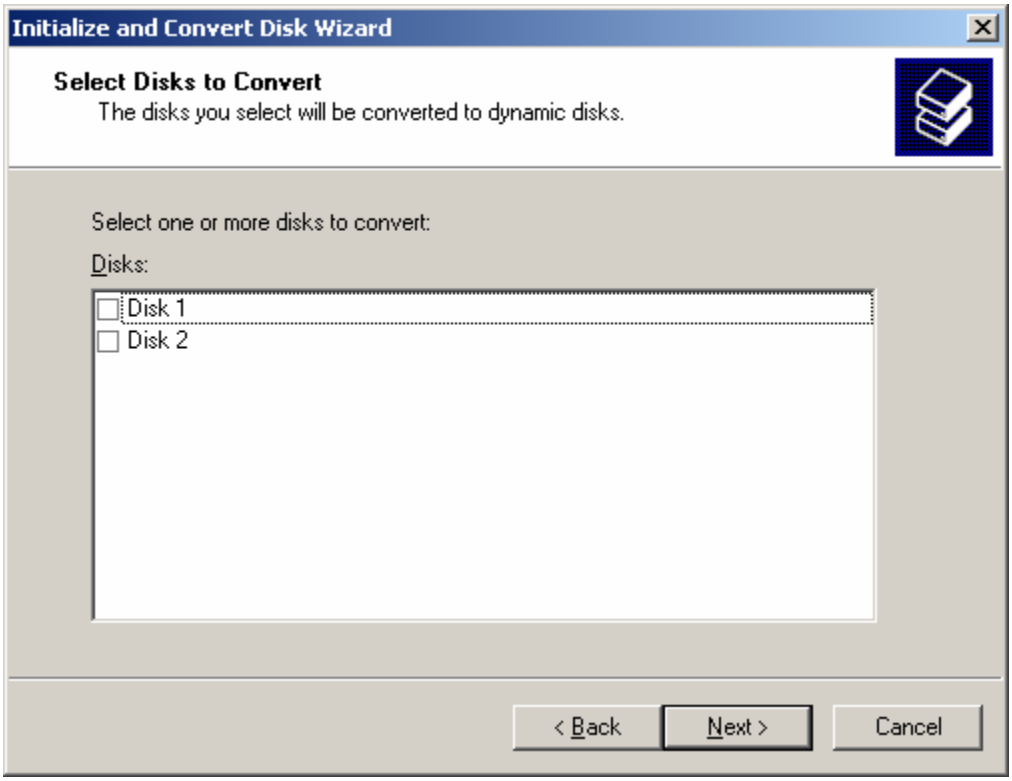

Figure 3.17: Ensure that iSCSI SAN volumes are not converted.

Once you have ensured that none of the new disks will be converted to dynamic disks, click Next. On the Completing the Initialize and Convert Disk Wizard screen, click Finish.

Figure 3.18 shows the Computer Management console, the Disk Management container, and the new iSCSI volumes. Disk 1 and 2 are the new iSCSI volumes that have been assigned to this Windows server. It is pretty easy to determine which disk is for which purpose because Disk 1 is 510MB and Disk 2 is 1.18GB.

The disk sizes are quite small and just for illustrating this example.

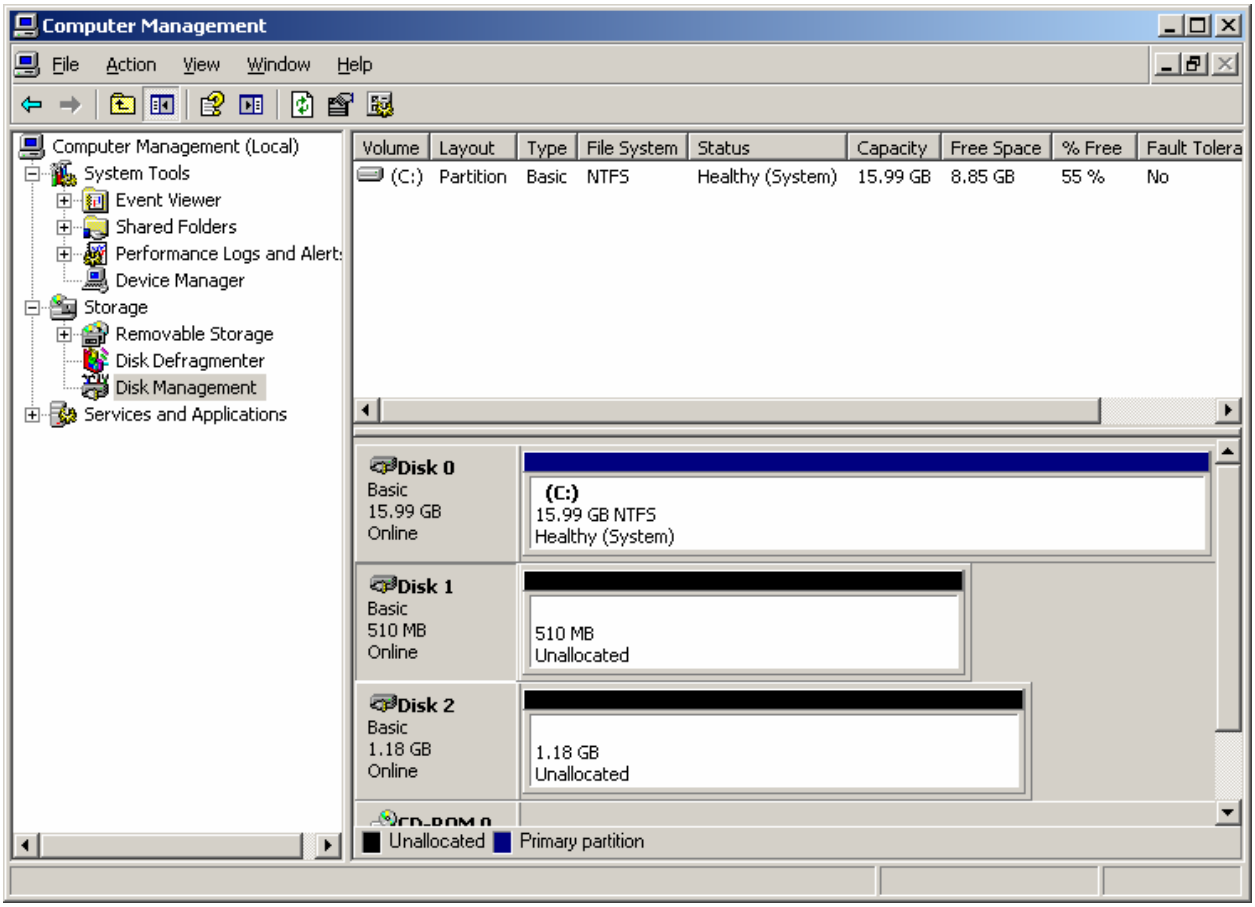

Figure 3.18: Viewing the new iSCSI disks.

We can match up each of the disks with the actual SCSI ID by selecting the disk drive (such as Disk 2), then right-clicking, and choosing Properties. This will show us the properties of the disk drive from the perspective of Windows. Notice that the location of the disk is Port(2) Bus 0, Target ID 1, LUN 0. This matches up with SAN volume SFEX01-MB01, which I had intended to be used for a mailbox database.

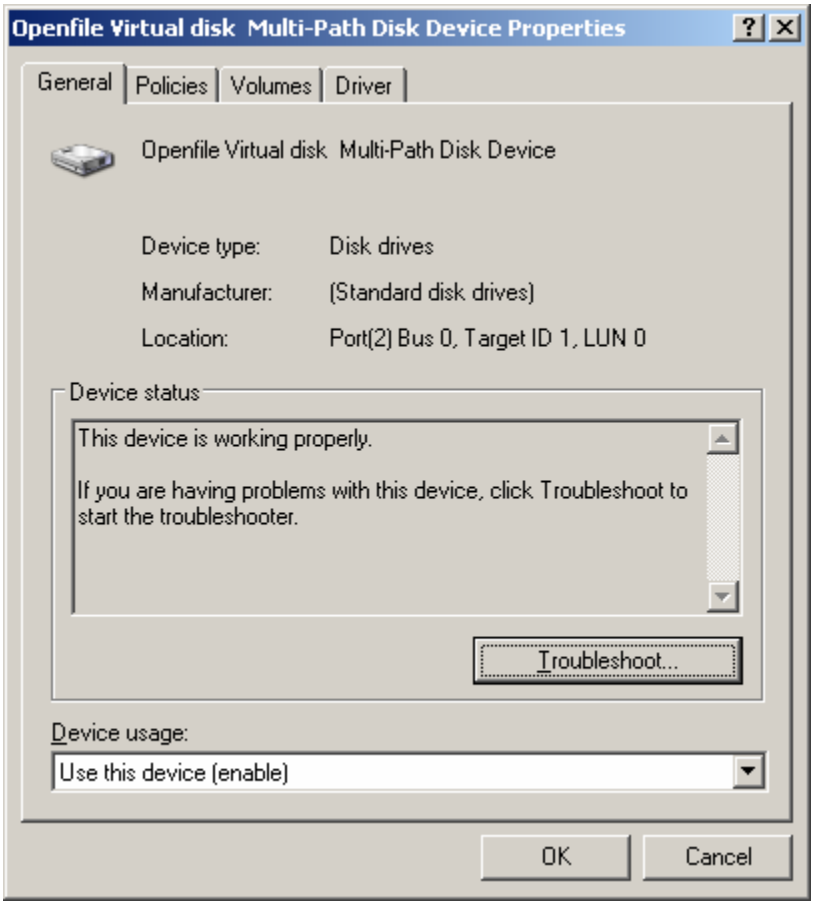

Figure 3.19: Properties of a disk using the Windows Disk Management console.

I now need to create a partition on each of these disks, assign the disks to a drive letter or mount point, and format the disks. On Exchange Server systems that require more than 20 volumes, you would assign the disks to mount points rather than drive letters; if you do not use mount points, you will run out of drive letters to assign to the volumes. To set up the disks, follow these instructions:

7. Right-click the unallocated space on each disk and choose New Partition.
8. On the Welcome to the New Partition Wizard page, click Next.
9. On the Select Partition Type page, select Primary Partition, and click Next.
10. On the Specify Partition Size page, ensure that the maximum size of the disk is being used for the partition. When finished, click Next.
11. On the Assign Drive Letter or Path page, specify the drive letter or configure a mount point. When finished, click Next.
12. On the Format Partition page, ensure that the NTFS file system is selected, that the allocation unit size is set to the Microsoft recommended value of 64K, and specify a useful volume label. When finished, click Next.
13. On the Completing the New Partition Wizard screen, click Finish. The partition will be created and formatted.

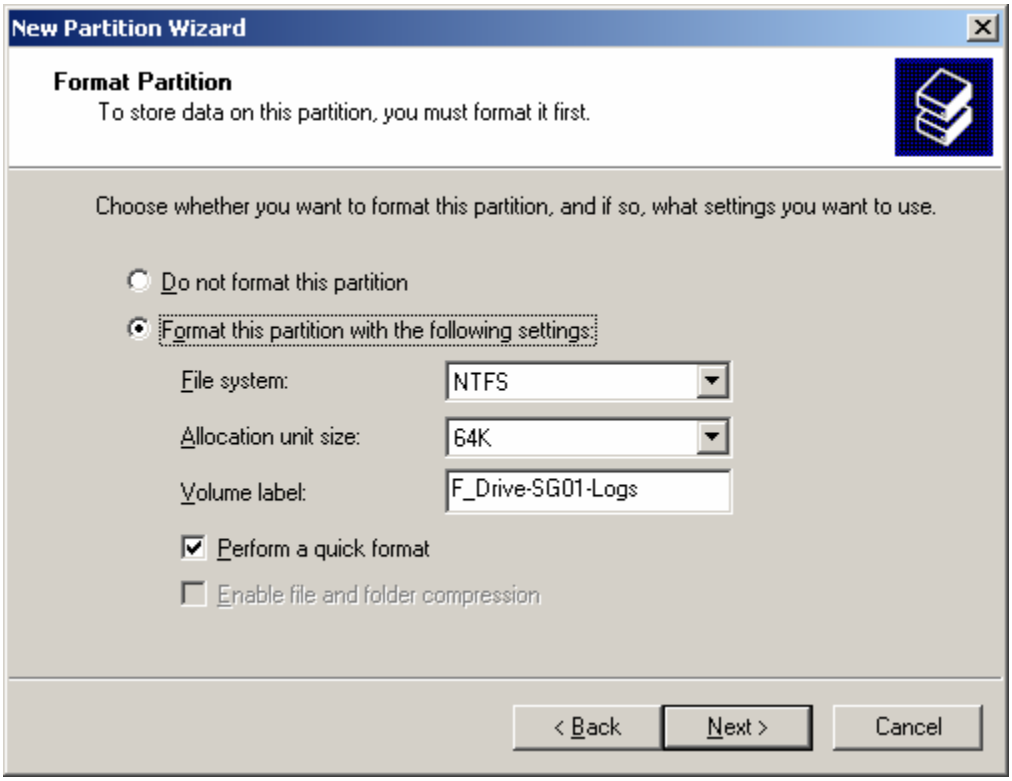

Figure 3.20: Specifying the file system type and allocation unit size.

These steps offer one procedure for partitioning and formatting a disk. Although this method is by far the simplest, you can also use the Windows 2003 SP1 utility diskpart.exe. The advantage of using diskpart.exe is that it allows you to create the partition so that it is evenly aligned with the 65th sector of the disk. I'll go into this in a little more detail in Chapter 4, but essentially the first 63 sectors of a disk are reserved for the boot sector (even if the disk is not bootable). This means that the first sector of the disk that is usable by partition will start on the 64th sector, but this means that the first logical sector spans multiple disk I/O boundaries.

For this reason, it is better to start the disk partition on the 65th sector rather than the first one available (usually the 64th); check with your SAN vendor to confirm that this is true for your storage system. In general, it is a good practice to create your partitions this way. Thus, we want the first partition to start on the 65th sector; because each sector is 512 bytes in size, the first sector should start at 32768 bytes.

To create a partition using this approach, open a command prompt and run diskpart.exe. This will offer the following output:

```
C:\>diskpart

Microsoft DiskPart version 5.2.3790.1830
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: SFOEX01

DISKPART>
```

Next, we want to list information about all the disks; I use the `list disk` command to do so:

```
DISKPART> list disk

  Disk ###  Status      Size     Free     Dyn  Gpt
  --------  ----------  -------  -------  ---  ---
  Disk 0    Online        16 GB  8033 KB
  Disk 1    Online       510 MB      0 B
  Disk 2    Online      1208 MB  1208 MB
```

Notice the output from the list disk command shows that for Disk 2, there is still 1208MB of free disk space. This is the disk that has not yet been partitioned or formatted. The Disk 2 referred to at the command line is the same Disk 2 we see in the Disk Management console, but you can confirm that it is the same physical disk using the `detail disk` command. Note that I must select the disk using the `select disk 2` command before I can use the detail disk command.

```
DISKPART> select disk 2
Disk 2 is now the selected disk.
DISKPART> detail disk
Openfile Virtual disk  Multi-Path Disk Device
Disk ID: 7739E0FB
Type   : iSCSI
Bus    : 0
Target : 1
LUN ID : 0
There are no volumes.
```

Notice that Disk 2 is Target 1, LUN ID 0. Now that the disk is selected, I can use the `create partition` command to create the partition. The align value is in KB, so I want to specify 32KB so that the first sector starts at 32,768 bytes (32 * 1024):

```
DISKPART> create partition primary align=32

DiskPart succeeded in creating the specified partition.
```

Once the partition is created, I can assign the partition a drive letter or a mount point and then format it. This can be done via the GUI.

> The allocation unit size for the disks should be 64KB when formatting the disks.

Finally, for each disk that I have created, I am going to create a text file that I call a marker file. The file name will have some information about the server to which the disk belongs, the local drive letter to which it should be assigned, and the SAN volume from which it came. For example, for the G drive that holds the transaction logs on server SFOEX01, the SAN from which the volume originated, and whose iSCSI SAN volume is SFEX01-LOGS01, I will name the file as follows:

```
SFOEX01_Drive-G_SF-SAN01_SFEX01-LOGS01.TXT
```

Let's break down the file name so that you can see the different parts:

```
Server name      SFOEX01
Disk drive Drive-G
SAN name   SF-SAN01
SAN volume SFEX01-LOG01
```

If the volume is a mount point instead of drive letter, you could substitute information about the mount point and where it is supposed to be found.

If you work with clusters or if you ever have to reconfigure your LUNs to reconnect to your Exchange Server, this will prove to be one of the most valuable 60 seconds you ever spent. The file does not have to contain any data, though you can certainly put some text in the file so that it is not 0KB. I also recommend that you mark the file as read-only so that an overanxious administrator does not delete it by accident.

Congratulations—your iSCSI SAN is now ready to use. However, this is by no means all that you would need to do for a successful deployment of iSCSI in your environment. The following list highlights additional steps that you might need to consider:

- Document your iSCSI configuration, including volumes and the initiators to which they are assigned
- Configure and schedule snapshot software for backing up data on the LUNs and configure the software to verify the integrity of Exchange databases, if applicable.
- Configure redundancy, fault tolerance, and/or multi-path I/O.
- Increase data security by enabling IPSec for authentication and/or data encryption.

For procedures on creating snapshots of Exchange data, configuring multi-path I/O, or IPSec, refer to your iSCSI SAN vendor's documentation or best practices for detailed information. These procedures will usually differ from one vendor to another.

Moving Exchange Databases and Logs to iSCSI LUNs

Once the iSCSI SAN volumes are ready for use on the Exchange Server, it is time to move data over to the volumes. This is a pretty straightforward task that most Exchange administrators will already be familiar with. The tasks and interface are slightly different for Exchange 2003 than for Exchange 2007, so I'll talk about them each briefly.

> ☞ Databases and transaction logs should be moved during periods of low usage and immediately after a backup so that you do not disrupt the users. Moving transaction logs immediately after a full or incremental backup ensures that there are not many transaction logs that must be moved.

Moving Exchange 2003 Data

Exchange 2003 databases and transaction logs are moved using the Exchange System Manager console program. It really does not matter the order in which you move the files. Each storage group's transaction log files are moved by displaying the properties of the storage group (see Figure 3.21). Use the Browse buttons to change the location of both the system path and transaction log locations.

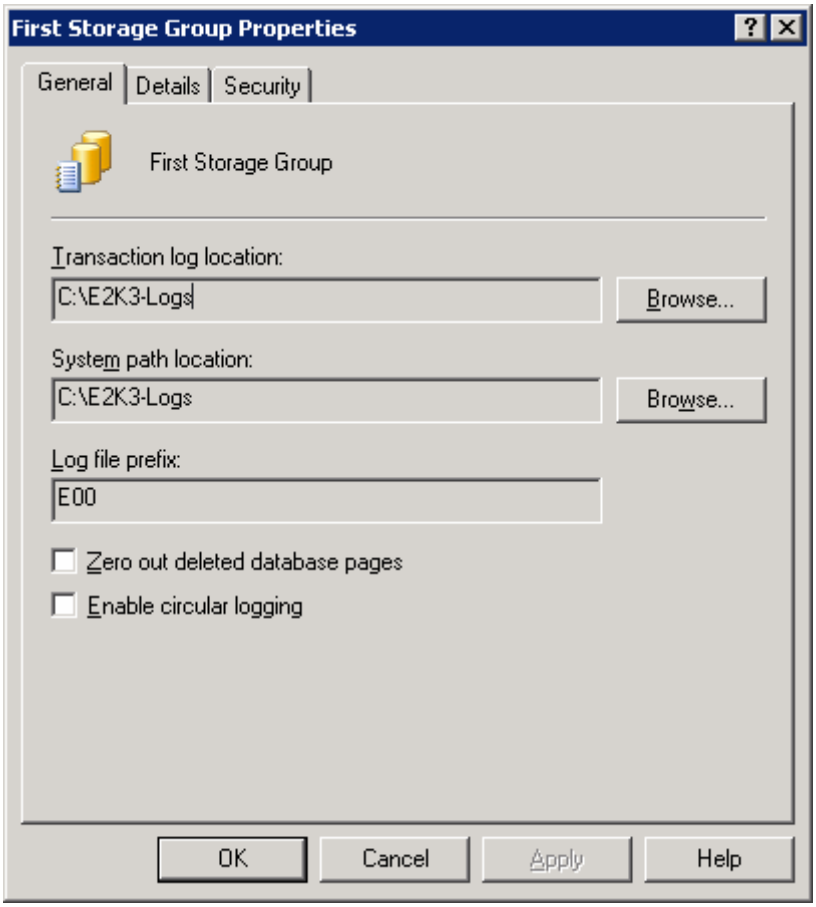

Figure 3.21: Moving a storage group's transaction log files.

When you change the location of the transaction logs or the system path, you will be warned that if you continue, all databases in this particular storage group will be dismounted. Any users using these databases will be disconnected. Depending on the number of logs to move, this may take a considerable amount of time.

Exchange 2003 stores are moved using the properties of the mailbox store. Figure 3.22 shows the Database property page of a mailbox store properties. From this screen, the administrator browses the available disks and specifies new locations for the native content database (EDB file) and the streaming database (STM file).

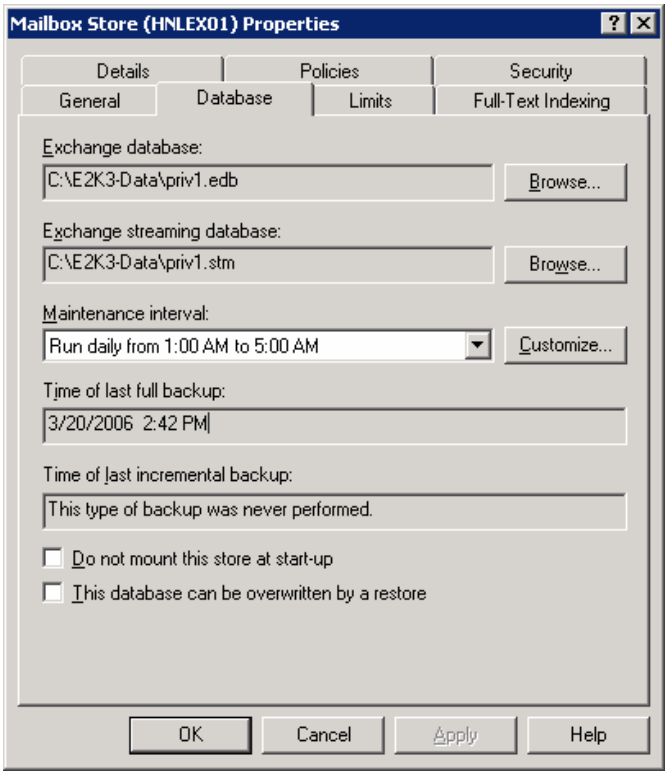

Figure 3.22: Moving the STM and EDB files.

Just like moving the transaction log files, when the administrator specifies the new location for the EDB and STM files and clicks Apply or OK, the store is dismounted, users are disconnected, and the files are moved. For larger database files, this task can take a considerable amount of time. Exchange databases should not be moved when doing so could interfere with user operations. If you work in a 24 × 7 shop, consider creating new mailbox stores on the new volumes and move the mailboxes over a few at a time.

> ☞ One interesting tip I have learned is that you can also rename the EDB and STM files when you move them. Doing so can be useful if you are trying to standardize your database file names or if you are not happy with how you originally named them.

Moving Exchange 2007 Data

The concepts behind moving Exchange 2007 databases and transaction logs are exactly the same as for Exchange 2003, but there are a few small differences. First is that the transaction logs in Exchange 2003 are 5MB in size while the transaction log files in Exchange 2007 are 1MB; the size was lowered to better accommodate the Exchange 2007 replication technologies such as local continuous replication (LCR), cluster continuous replication (CCR), and standby continuous replication (SCR). The smaller transaction logs do not change the actual volume of transaction logs, this just means there are more of them.

Another difference is that Exchange 2007 does not have the streaming database file, just a native content file (EDB file). The databases are just referred to now as 'databases' rather than stores.

Finally, we have an additional option for moving database files and transaction logs. We can either use the GUI or the command-line interface. Let's look at both methods for moving the files. First is the Exchange Management Console (EMC); this is the GUI for Exchange 2007.

> ☞ Prior to moving Exchange 2007 databases or transaction logs, suspend local continuous replication if you have implemented it.

First we need to load the EMC and navigate to the Server work center and then to the Mailbox subcontainer. This is where we will find the Mailbox servers and thus the storage groups and databases. Locate the storage group whose files you want to move to another volume and select it. You can either right-click it and select Move Storage Group Path or you can select that option from the Actions pane of the EMC. Doing either launches the Move Storage Group Path wizard (see Figure 3.23). Make sure that you select to move both the log files and the system files path. When you are ready, click Move.

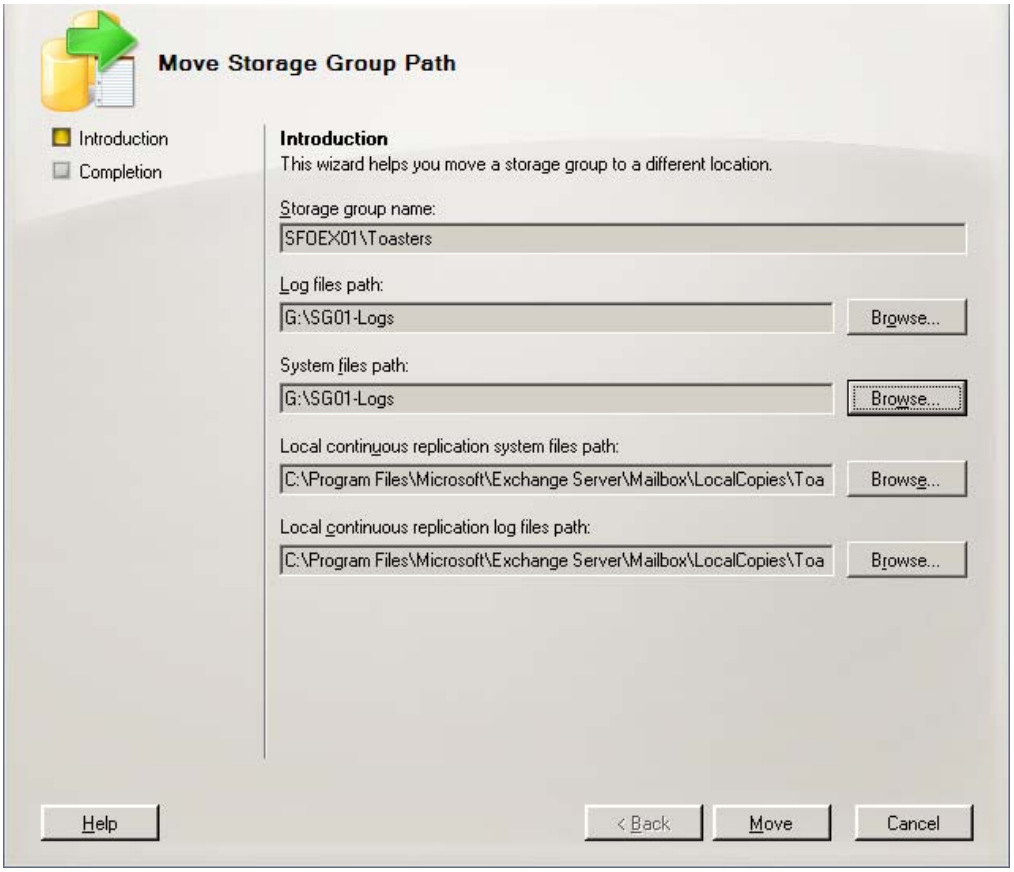

Figure 3.23: Moving Exchange 2007 storage group files.

Once the wizard has completed, you will notice that the Completion screen includes the Exchange Management Shell (EMS) command that was used to move the log files and system files. The following is the command used to move the Toasters storage group files to the G:\SG01-Logs folder:

```
Move-StorageGroupPath -Identity 'SFOEX01\Toasters' -LogFolderPath
'G:\SG01-Logs' -SystemFolderPath 'G:\SG01-Logs'
```

Next, we want to concern ourselves with moving the database file. Highlight the database in the EMC, and select Move Database Path task from the Action pane or from the context menu. This will display the Move Database Path wizard (see Figure 3.24). Select the new location for the database file, then click Move. The move process will kick off any users that are connected to this database; the move could take a considerable amount of time if the database file is large.

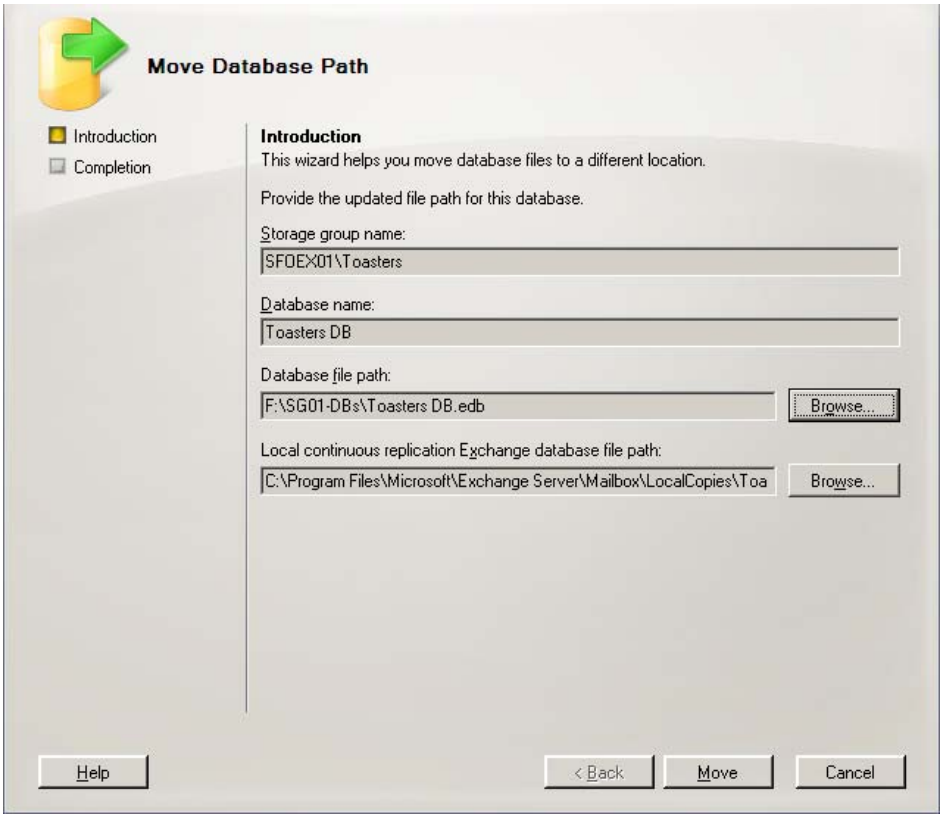

Figure 3.24: Moving an Exchange 2007 database file.

When the move is completed, the Completion page includes the EMS command that was used to move the database. In this case, to move the Toasters DB to the F:\SG01-DBs folder, the following command was used:

```
Move-DatabasePath -Identity 'SFOEX01\Toasters\Toasters DB' -
EdbFilePath 'F:\SG01-DBs\Toasters DB.edb' -CopyEdbFilePath
'C:\Program Files\Microsoft\Exchange
Server\Mailbox\LocalCopies\Toasters\Toasters DB.edb'
```

Summary

This chapter discussed the basics of iSCSI and how to implement a basic iSCSI SAN using Windows 2003, the Microsoft iSCSI Initiator software, and Exchange Server. The procedures for doing this are fairly simple. There are few important pieces of information to keep in mind. First and foremost, you want to make sure you have the latest updates to the Windows OS, including the newest version of the iSCSI initiator software.

Second, you want to ensure that you are precisely following any guidance provided to you by your iSCSI SAN vendors. Best practices for configuring the iSCSI initiator (and other hardware) will vary from one vendor to another, so make sure you are using the practices that are best for your hardware and software.

This chapter covered some of the most basic steps necessary to install and configure the iSCSI initiator so that you can set up a basic iSCSI SAN. The examples that I completed used an opensource OpenFiler virtual machine, which is ideal for practicing and refining your iSCSI skills. However, I did not cover more advanced configuration options such as implementing multi-path I/O, snapshots, or IPSec.

Chapter 4 will cover some of the generic best practices you should follow when implementing an iSCSI SAN. For the most part, these should transcend any particular vendor's requirements or specific recommendations.