



realtimepublishers.comtm

The Definitive Guidetm To

Windows 2003 Storage Resource Management

VERITAS[™]

Evan Morris

Chapter 4: Developing the Storage Resource Management Solution.....	90
Phase 3: Developing the SRM Solution.....	90
SRM Project Team Roles.....	91
General Project Guidelines	92
Project Tools	92
Deployment Template.....	92
Communication Plan.....	94
Minimizing Disruptions	95
Deployment Topology	95
Delegating Administrative Functions	95
Developing an Organizational Storage Policy	96
Product Support and Escalation Procedures	97
SRM Goals.....	97
SRM Tools	97
Windows Server Disk Quotas.....	97
Creating Additional Storage Capacity	103
Windows Server DFS	104
Windows Server RSS.....	104
Supported Storage Systems.....	105
Networked Storage.....	105
Virtualization	107
Storage Solutions	108
Storage Service Provider.....	108
Storage Devices	108
Disk Drives	110
Storage Density.....	110
SCSI versus IDE	111
RAID.....	113
RAID Controllers.....	114
Performance Design.....	115
Performance Design: Exchange Server Example	117
Summary	118

Copyright Statement

© 2004 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

Chapter 4: Developing the Storage Resource Management Solution

In the previous chapter, we took a look at capacity planning to ensure that you don't underestimate storage resources' growth and the resulting impact on IT administrative capabilities. I gave an example of using an SRM tool to analyze storage at more than one point in time so that you can perform trend analysis. I covered storage resource planning, and explained how to plan a course of action with a focus on increasing storage capacity or improving performance. Then I illustrated how you can use an SRM tool to support your decision to more efficiently use your existing storage rather than deploy new storage. We'll dive into more detail about this decision as we develop an SRM solution for your organization.

In this chapter, we'll explore two broad areas: structuring the storage management project and using storage management tools to make better use of your existing storage or your newly deployed storage. To address the need to expand storage arrays, you can attach to WS2K3 DFS; I'll discuss this option and the hardware-selection process involved in migrating to new storage systems. In addition, we'll take a brief look at the Windows Server RSS to decide whether it's right for you. Finally, to illustrate how to design with performance in mind, we'll look at hardware and a performance design example for Windows Server that uses Exchange Server 2003 as a storage application.



A word on methodology: As stated in the first chapter, I've organized this book in a project methodology based on the MSF. Because the MSF was created for software development or programming, the structure is slightly different than for IT infrastructure solutions or projects. This difference was reflected in a recent update to the Microsoft solutions architecture; however, I prefer to stick to the original MSF because it calls out the development and testing (pilot) portions of the project with greater emphasis. This point is important, as greater emphasis is often placed on beta testing software development solutions rather than infrastructure solutions. The net result of more beta testing is higher-quality solutions, as you learn about and change your design to better fit the business workflow.

Phase 3: Developing the SRM Solution

Table 4.1 shows Phase 3 in the overall SRM deployment methodology.

Phase	Process	SRM
Development	Build the solution (for testing)	Use the storage reporting tools and prepare to move them to real-time production usage.

Table 4.1: Phase 3 of the SRM deployment methodology.


Before we look at what will be accomplished in this phase and how, you'll need to decide who will be performing which type of work.

SRM Project Team Roles

As I break out each of these roles, don't be surprised if you fill many of the roles. In a small business environment, one person often fulfills all or most of the duties. In a larger or enterprise environment, identifying the duties that one person could perform (given infinite resources such as time and leadership charisma) and offloading duties to other individuals becomes an essential task. By no means does Table 4.2 imply that each role requires a separate person or that one person couldn't handle multiple roles. The number of people involved simply depends on the scope or size of the project (for example, whether you will build a new SAN) and the resources available.

SRM Project Team Role	Function
Project Sponsor	Provides funding and overall go or no-go decisions. Narrows the vision (what could be accomplished) to the scope (what will be accomplished). Defines which problems will be solved.
Project Manager	Allocates resources (people, tasks, funding, and so on). Manages the project scope, ensuring that the tasks performed and resources used are targeted at successful project completion.
Technical Architect	Designs the solution to be implemented and defines how problems will be solved.
Technical Advisors	May be called in on short notice to provide guidance about which direction to proceed or how to repair a specific problem.
Subject Matter Experts	May be called in on short notice to provide guidance about how to repair a specific problem.
Design and Build Engineers	Make decisions about the selection and installation of SRM products. Performs the installations or creates the process to delegate the installations to regional administrators or server operators.
Test Engineers	If the team is large enough, separate the test engineers from the design engineers because the test engineers lend a different perspective about how the solution will be used; might consist of pilot users drawn from the company or organization.
Communications Specialists	Communicate the goals and benefits of the project to the business or end-user community. This duty is a specific role in larger organizations and is mainly an issue when the enterprise has an existing channel in place (such as Human Resources); otherwise the project manager or team may fulfill this role.
Regional Administrators	Perform the same functional responsibility as the design and build engineers, but might not be able to participate in the centralized project in a distributed environment; might apply the architecture to their regional systems and provide feedback to the centralized team.
Server Operators	Perform some installations and routine maintenance. Can provide feedback about the quality of project documentation, such as operations guides.

Table 4.2: SRM project team roles.

 A note about technical advisors and the wisest use of consulting resources: Doesn't the word *consult* mean that you are asking for advice? Ideally, this person is someone whose experience you can draw upon. They are *consultants*, not employees—they should tell you how to do your job (or how to do it better), not actually do your work for you. This role in the project should be limited in duration and scope, and if the prospective consultant is not willing to work in this manner, they're looking for long-term employment!

General Project Guidelines

These project guidelines are based on lessons learned from previous projects. Regardless of your end goals and the means to accomplish them (which we identified in Chapter 3), these are the principals that will hold true across the widest variety of projects.

Project Tools

In Chapter 3, we explored how to make SRM decisions such as which options or choices to make to better utilize existing capacity or to expand capacity. These ideas were central to the themes covered in that chapter. Similarly, in this chapter, the deployment template that Table 4.3 shows plays an instrumental role and is the focal point for this chapter. This tool is meant to be a project aid for your SRM deployment and will serve you well printed out and pinned to your cubicle wall or distributed to management at that all-important meeting.

Deployment Template

Table 4.3 outlines a deployment template that you can customize to suit your project. You may also use it for communication to upper management, as it outlines the SRM goals, processes, benefits, and status. I have designed it to be used as a template for your project, meaning that you can customize each item for what you plan to accomplish, removing items that may not be immediately relevant.

Goal	Process and End Result	Priority and Estimated Benefit	Team Members	Status
Delegate administrative functions; increase end-user knowledge and awareness	Create and evaluate communication channels, conduct regular project meetings, and train the involved parties in the SRM solution and administrative duties.	1. Increases storage-consumer buy-in and reduces total cost of ownership (TCO).		Get feedback on customized reports. Test communication channels such as email and corporate intranet/portal as well as traditional paper-based means (brochures).
Match storage location alternative to type of information	Evaluate the storage location alternatives (print out the table from Chapter 3).	2. Reduces administrative overhead and increases functionality to business users, such as customer responsiveness.	(Insert the names of the team members responsible for this task.)	Identify storage location alternatives and provide best-matched storage options.

Goal	Process and End Result	Priority and Estimated Benefit	Team Members	Status
Improve system availability and reduce backup and recovery windows	Separate business-critical storage from non-essential storage.	3. Reduces administrative overhead of maintenance and recovery operations.		Requires a combination of the other solutions detailed below.
Eliminate unused/ orphan files	Perform SRM reporting and administrative cleanup.	4. Reduces wasted space on file servers and application servers.		As simple as a weekend cleanup job (assuming you have the right tools).
Eliminate unnecessary files (based on type)	Implement file blocking.	5. Reduces wasted space on file servers and application servers.		Evaluate and test SRM products.
Reduce storage consumption (that is, reduce non-essential files)	Implement file system quotas per user or group.	6. Encourages users to maintain appropriate levels of information storage.		Evaluate and test SRM products.
Eliminate duplicate files	Perform SRM reporting and administrative cleanup. In addition, this step requires measures to ensure prevention of future file duplication (involves giving users better ability to share files).	7. Reduces wasted space on file servers and application servers.		Create shared file mechanisms (common storage or collaborative applications), run duplicate file reports, and migrate duplicate files.
Eliminate unnecessary files (based on aging)	Use HSM.	8. Offload infrequently used files.		Decide on date criteria, select an HSM product (native Windows Server versus a third-party), and evaluate a physical storage medium.
Expand existing storage for well-known \\Server\shares (part of mapped drives for logon scripts)	Create DFS architecture.	9. Consolidate existing servers and storage into a unified namespace.		Pilot in lab to test replication over WAN.

Goal	Process and End Result	Priority and Estimated Benefit	Team Members	Status
Expand existing storage (as previously mentioned)	Expand storage arrays (more physical drives).	Low priority; requires the ability to add to storage capacity; few solutions support online capacity expansion (see OVG and VVM in Chapter 3).		Test volume mount points versus expanding existing arrays (if this testing is possible, it usually requires downtime).
Expand existing storage (as previously mentioned)	Replace servers or storage.	Low priority; requires additional funding and cost-benefit analysis.		Investigate DAS, NAS, and SAN alternatives, both the initial cost and as an ongoing management solution.
Improve existing storage performance	Replace or upgrade servers or storage devices.	Low priority; requires additional funding and cost-benefit analysis.		Investigate DAS, NAS, and SAN alternatives, both the initial cost and as an ongoing management solution.

Table 4.3: Deployment template outlining the SRM goals, processes, benefits, and status.

Communication Plan

An essential part of an SRM project is identifying who needs to know what and how. For example, you must develop a process for communicating to your upper management the ongoing status and progress of the project. Have you ever lifted your head from what you are working on to realize the entire day is gone, and you have come nowhere near accomplishing what you originally set out to accomplish? Where did the day go? What did you get done? And most importantly, how will you communicate this day in your status report to management? Looking back over the day, why did your immediate focus shift to the new task, and did it move your project closer to the goal? This type of information needs to be communicated not so much on a daily basis but more likely on a weekly basis: Is the project moving closer to the originally stated goal, and are resources being used in the right manner?

Table 4.3 is provided as part of developing your communication plan, and can be used to communicate to upper management the individual goals, benefits, and the status of your SRM project.

Minimizing Disruptions

At some point in the project, you will make tradeoff decisions between doing an in-place upgrade of an existing storage system and building a new storage system and migrating files to it. An analogy for doing an in-place upgrade of an existing storage system versus migrating to a new one is whether you should remodel your existing home or move to a new one. You know the hassles of moving, but the reason you face those hassles is to get a bigger, better home. But if you cannot afford that, the piecemeal approach is to add on to your existing house as much as you can afford at the time. Just as in remodeling a home, the in-place storage upgrade can be as disruptive as moving to a new home.

Deployment Topology

If you do not have a clear picture or diagram of the topology of your organization, you will need to develop one as part of the SRM project. You will need to map out the storage topology of the organization to reflect both the geography and the administrative model, whether it is centralized or distributed. Your organization may exist primarily in one or a few locations, making for a centralized model, or it may have branch or division offices spread across the world that are maintained autonomously. The hierarchy of the deployment model—centralized or distributed—will have an influence on the design of your SRM solution and project. For example, in the distributed model, the decisions that you make as the top-level administrator may or may not be accepted by the administrators at the distributed locations. Perhaps they will not have administrative override and must live with your decisions. Either way, you need to consider the topology as part of testing the SRM solution because the solution may work well in the main office data center but break down at the remote offices.

Delegating Administrative Functions

The benefit to the organization of delegating administrative functions is twofold. First, delegating administrative functions is cost-effective if the SRM solution can be operated and maintained by lower-cost employees. The more employees that are able to use the system, the greater the chance of its survival in the long-run, as the organization will never fall into the trap of losing the only employee who knows how to operate the system. Second, as you increase end-user knowledge and awareness of the SRM solution, you increase the end-user buy-in that this solution isn't just something that is imposed from on high but a tool that is designed for their benefit. In the case of managing files and important business information, the capabilities and decisions must absolutely be distributed to the information owners and end users of the files, as there is no way that the storage systems administrators can know the relative importance of each and every file.

Developing an Organizational Storage Policy

Although a plan of action for storage is certainly important, it must be married with an organizational storage policy. As we develop our technological solutions for managing storage, they will be used to support the organizational storage policy that we will develop in the next phase of the project. In fact, the two must complement each other, as a conflict between storage design and organizational storage policy is disruptive to business process. For example, if user space is limited by policy on a business-critical storage system, but the tools are not in place to enforce it, additional administrative burden is created to monitor and maintain the system.

The end result of this twofold strategy is to increase the availability of storage systems while reducing the cost of administrative maintenance. These are difficult goals to attain. As we increase the importance and usage of storage systems, we must also increase their fault tolerance and ability to recover from any disruption to service. In the area of TCO, entropy is our enemy—if a system is left alone over time, it will begin to decay. For storage management, this idea means that if the tools are not in place to monitor and maintain the system in an efficient manner, the state of storage will move towards a natural decay in order and chaos. Perhaps you have seen the results of this corrosion for a particular file server: an ever-increasing number of files being stored, with diminishing access patterns, using ever more storage, which increases the backup and recovery window. In the next section, we will look at the impact of increasing our storage capacity.



Implementing storage policies without the aid of your compliance department can be difficult. Many corporations have very specific rules about what can and should be kept and for how long. These rules are often the first place to start in developing a policy, and if a policy is developed without checking with the compliance department first, you can get into a situation in which the policy that the IT department decides on isn't backed up by a corporate policy.

One more point about the backup and recovery window, as the concept of SLAs can have a powerful impact in our storage capacity planning. When users are storing more and more information, throwing storage space at the problem is a temporary solution at best. Before long, the end result is that business-critical or even mission-critical data can be compromised because it is mixed with files that are no longer essential, and all the data must be backed up or recovered during the same time period or window. This behavior is the driving force behind HSM—moving non-essential files away from business-critical and mission-critical information. The SRM plan can include how HSM will be incorporated, and the benefit to the company (which drives the cost justification) is the improved response and protection of business-critical and mission-critical information.

Product Support and Escalation Procedures

As part of the assignment of roles in the project, an ongoing role will be that of supporting and maintaining the SRM solution, including dealing with any end-user issues that will arise. In the next section, we will dig into SRM tools, so carry forward this reminder that product support and escalation needs to be laid out before you roll out the SRM solution; otherwise you may stress your Help desk and support system. For example, when we look at setting disk quotas on end users, consider the wording of the notification message that users will receive. Ideally, it will not create any confusion, and should give a point of reference that the user can go to get more information, such as a Web page or a Help desk phone number.

SRM Goals

As we dig into SRM policies and the tools used to create them, keep in mind the four goals of SRM:

1. Eliminate duplicate files (and improve the sharing of files)
2. Eliminate unused files (based on aging and orphaned files)
3. Eliminate wasted space (from non-essential files, based on file type)
4. Reduce storage consumption (by setting disk quotas)

We will translate these SRM components into five distinct actions in our development. Five actions rather than four because eliminating unused files consists of cleaning up two different types of files that are no longer used— orphan files and aged files—and they need to be identified separately.

SRM Tools

In this section, we will see how the core Windows Server functionality and features, including quota management, can attempt to solve the SRM goals. After we have exhausted the capabilities of Windows Server, we will turn to a more comprehensive third-party solution.

Windows Server Disk Quotas

Regarding the goals of eliminating duplicate files, eliminating unused files (aged and orphans), eliminating wasted space, and reducing excess consumption, the core Windows Server functionality is limited to the Disk Quota feature. (I covered some limited reporting functionality in Chapter 3 and introduced disk quotas in Chapter 1).

You access the Windows Server disk quota feature by right-clicking a disk volume (usually synonymous with a logical drive that has a letter such as D) in Windows Explorer, and selecting the Quota tab, as Figure 4.1 shows. In the example that the figure illustrates, I have enabled quota management and set several of the options that are not set by default. For example, I have set the warning level at 190MB and prevented users from writing any further than 200MB. I have selected the property so that when users exceed their disk space, the system will log an event on the server. Because I selected the *Deny disk space to users exceeding quota limit* check box, the quota limit is a hard quota. If I did not select this option, the quota limit would be a soft quota, which is used more for informal reporting and alerting end users as well as administrators of disk-usage levels.

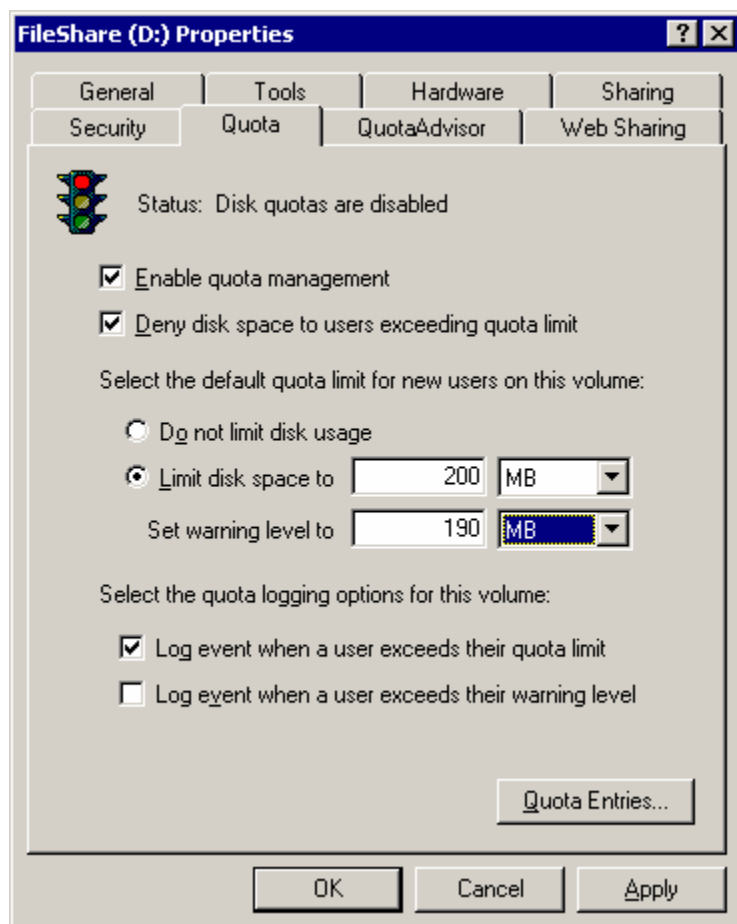



Figure 4.1: Properties for a Windows Server volume showing quota-management settings.

 Disk quota notifications are not written to the event logs immediately; they are written to the event logs every hour. To modify the default one-hour disk quota notification time, locate the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem registry key, and create a new REG_DWORD type entry called NtfsQuotaNotifyRate. Set the value of this entry to the desired interval (specified in seconds).

There are a few things to consider about Windows Server disk quotas before you rush to implement them:

- Windows Server disk quotas are based on disk volumes and cannot be set on the level of individual objects files or folders. (This limitation doesn't exist in more robust SRM products. Many such products let you specify that a folder does not grow to, for example, more than 500MB, regardless of which user is placing the files in that folder, including anonymous or guest accounts). Windows Server disk quotas are assigned to users and group members.

- Windows Server disk quotas are based on file ownership. This setup is generally true of any disk quota system or software, as it helps to fairly assign disk usage based on who is using or storing the files. The one exception is that members of the Administrators group (either at the local server level or at the domain level, depending on the server's domain membership) are not assigned individual file ownership. This exception makes applying disk quotas to administrators difficult, as file ownership is assigned to the Administrators group instead of each logon account.
- The Quota tab is not displayed on the volume properties unless that volume is formatted with NTFS and the logged on user is a member of the Administrators group for that machine.
- Windows Server disk quotas are based on uncompressed file sizes, meaning that the end users cannot increase the amount of available space by compressing the data.

When you enable disk quotas and click OK or Apply, you will see the pop-up warning that Figure 4.2 shows. The message informs you that the disk quotas will not take effect immediately.

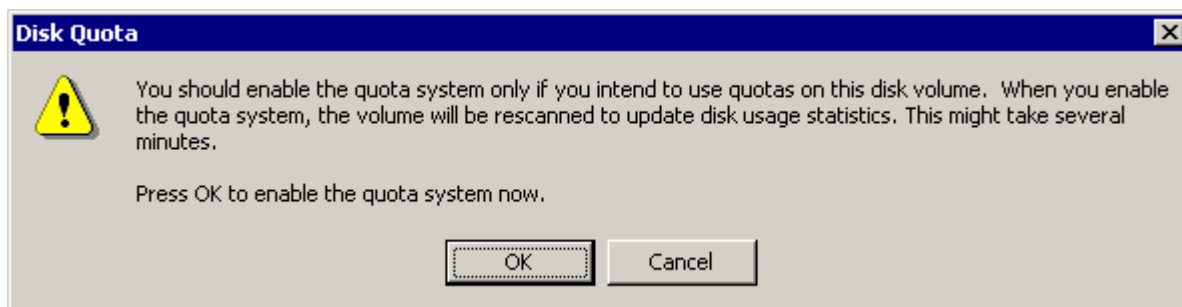



Figure 4.2: Pop-up warning when you enable disk quotas.

Notice that the quota settings specified in Figure 4.1 are for new users on the volume. To selectively control individual user settings, use Quota Entries, as Figure 4.3 shows. When you click Quota Entries on the Quota tab, the Quota Entries table will show you any existing quotas and allow you to add user-specific quotas by selecting New Quota Entry from the Quota menu.

 The system takes a very long time to fill the table; you will see the [Retrieving Name] entry for quite some time, as I have found this report to be slow, even on the faster systems in my company.

Status	Name	Logon Name	Amount Used	Quota Limit	Warning Level	Percent Used
Above Limit	[Account Information Unav...	S-1-5-21-606747...	13.31 GB	4.88 GB	4.39 GB	272
Warning	Alice	SANFILE\Alice	535.56 MB	40 GB	40 KB	1
Warning	MikeS	SANFILE\MikeS	8.11 GB	40 GB	40 KB	20
Warning	Tran	SANFILE\Tran	4.19 MB	40 GB	40 KB	0
Warning	Larry	SANFILE\Larry	39.82 MB	40 GB	40 KB	0
Warning	Liz	SANFILE\Liz	223.37 MB	40 GB	40 KB	0
Warning	Carol	SANFILE\Carol	1.1 GB	40 GB	40 KB	2
Warning	Jane	SANFILE\Jane	20.17 MB	40 GB	40 KB	0
Warning	BobE	SANFILE\BobE	5.28 GB	40 GB	40 KB	13
Warning	MikeZ	SANFILE\MikeZ	39.87 MB	40 GB	40 KB	0
Warning	Robert	SANFILE\Robert	207.21 MB	40 GB	40 KB	0
Warning	Ted	SANFILE\Ted	92.51 MB	40 GB	40 KB	0
Warning	Mary	SANFILE\Mary	9.3 MB	40 GB	40 KB	0
Warning	Jan	SANFILE\Jan	173.87 MB	40 GB	40 KB	0
Warning	George	SANFILE\George	22.14 MB	40 GB	40 KB	0
OK		BUILTIN\Administ...	12.52 GB	No Limit	No Limit	N/A
OK		NT AUTHORITY\...	11.81 MB	4.88 GB	4.39 GB	0
OK	[Account Information Unav...	S-1-5-21-606747...	3 KB	4.88 GB	4.39 GB	0

18 total item(s), 1 selected.

Figure 4.3: The Quota Entries table for Windows Server disk quotas on the D volume.

Figure 4.4 shows the Add New Quota Entry dialog box that you see when adding multiple users to the disk quota on a volume. Note that you can set this disk quota at different levels than the primary disk quota for new users that you set on the Quota property page (which Figure 4.1 shows).

Add New Quota Entry

User: < Multiple >

Set the quota limit for the selected user(s):

Do not limit disk usage

Limit disk space to

Set warning level to

OK Cancel

Figure 4.4: The dialog box for adding multiple users to a Windows Server disk quota on a volume.

Any time that you make a change to the quota settings, you will need to wait for the Quota Entries table to refresh. The table is so slow to retrieve information that I have often seen it only do 2 or 3 users per minute. Thus, I advise leaving the console open or you will suffer this painful refresh period every time you launch the Quota Entries table. However, the table isn't updated in real time, so eventually you will have to hit refresh.

Using this information in a report is quite easy. To drag-and-drop information from the Quota Entries table to a document or spreadsheet, open a Word document or Excel spreadsheet while the Quota Entries table is building. When it is done, select the Quota Entries you want to include in your report, and drag the rows to the program you are using to create the report. You might need to drag and hold the selection on the taskbar until the program pops up to the foreground. At this point, you can apply formatting, as Figure 4.5 shows, and save the file.

Quota Entries for E_NTFS_280GB (E:)						
Status	Name	Logon Name	Amount Used (MB)	Quota Limit (MB)	Warning Level (MB)	Percent Used
OK		BUILTIN\Administrators	17,311	No Limit	No Limit	N/A
OK		W2KFS017\Alice	-	30	20	-
OK		W2KFS017\Aram	2	30	20	7
OK		W2KFS017\Armitage	9	30	20	29
OK		W2KFS017\Ashcroft	5	30	20	17
Warning		W2KFS017\Balmerino	22	30	20	74
Warning		W2KFS017\Barbour	21	30	20	69
Above Limit		W2KFS017\Bartelano	32	30	20	105
Above Limit		W2KFS017\Barthelme	30	30	20	100
Above Limit		W2KFS017\Bates	30	30	20	100
Above Limit		W2KFS017\Beale	34	30	20	112
OK		W2KFS017\Bellingham	-	30	20	-
Above Limit		W2KFS017\Bishop	34	30	20	112
Above Limit		W2KFS017\Blakesley	33	30	20	110
Above Limit		W2KFS017\Bland	33	30	20	110
Above Limit		W2KFS017\Bob	76	30	20	253
Above Limit		W2KFS017\Bonsall	35	30	20	115
Above Limit		W2KFS017\Bousfield	35	30	20	115
OK		NT AUTHORITY\SYSTEM	1	200	180	-

Figure 4.5: Windows Server Quota Entries report in an Excel spreadsheet.

Settings that you create for user quota entries can even be copied from one server to another by using the quota export and import functionality. Just set up your quotas, and export to a file (the system won't suggest a file extension to use, so you will have to come up with one such as .UQE). Then start the Quota Entries report for the volume to set quotas on, and import the file.

User quota entries cannot be deleted until all files on the volume owned by the user are either deleted or moved to another volume. You will be prompted to delete, take ownership of, or move the files as Figure 4.6 shows. So be careful before applying the quotas, as you may have to raise the disk usage limit instead of deleting the user quota. Also note that you can only delete files and not folders through this interface.

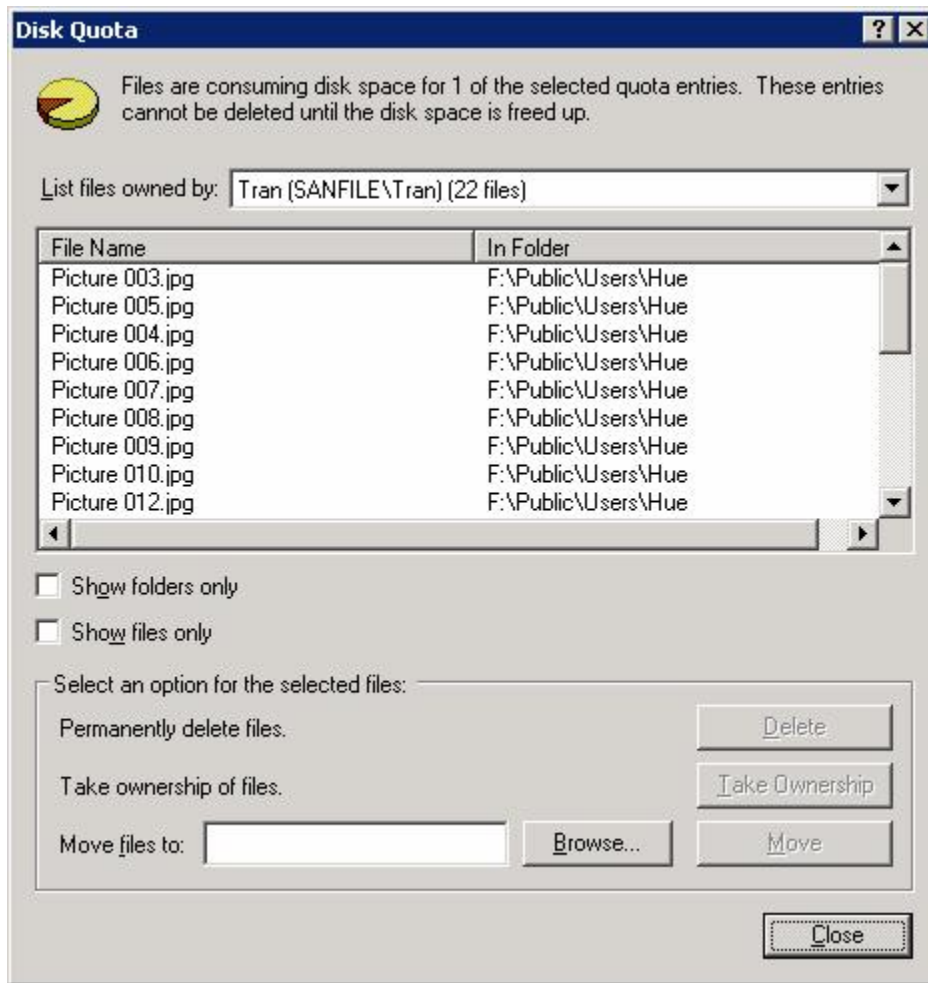


Figure 4.6: Prompt to delete, take ownership of, or move files when deleting user quota entries.

A final caution about using Windows disk quotas—when implemented, they can give a disk the appearance of being limited in size, as Figure 4.7 shows. This appearance can cause some confusion, especially if a new administrator on the system does not realize why the disk size displays incorrectly and why the free space is so low—these are both calculated by the quota. The quota limited size even shows in some disk utilities, such as the defragmenter, which then incorrectly calculates the percentage of free space.

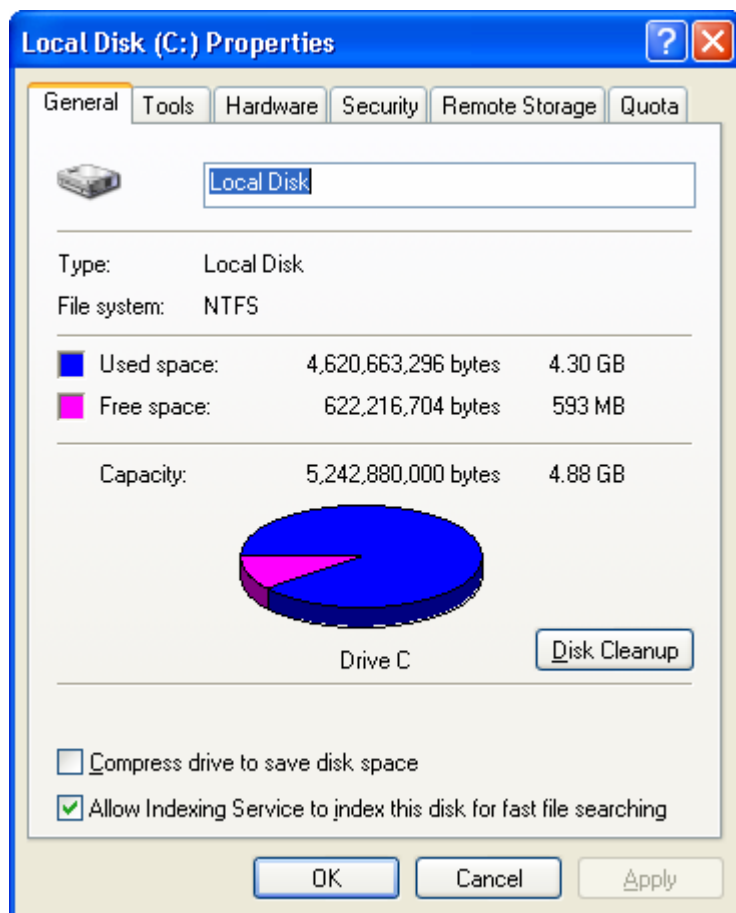


Figure 4.7: Disk Quota limits appearance of disk size.

Creating Additional Storage Capacity

When you decide to create additional storage capacity, you have three choices: you can expand an existing array, migrate to a new storage system, or attach to another storage system using the Windows Server DFS. We'll look at each of these options in turn.

Expanding an existing array may or may not be possible depending on several criteria. First, if using DAS, the physical cabinet boundaries may be a constraint. If you're unable to add drives, the alternative may be to pull the existing drives and recreate the array using larger capacity drives. To retain the same drive in the OS, you'll most likely need to create the array and restore the data from backup. If adding more drives to the storage cabinet is a possibility, the methods to expand the array (the physical drive presented to the OS) must be considered. The methods will depend on what is available to you. For example, your method will be different if you're using dynamic disks or a full-featured third-party management tool.

Windows Server DFS


You can use the Windows Server DFS to expand storage on an existing server by creating a logical view of the storage that is actually distributed across multiple disks or servers. When an application or a user attempts to access a data resource, the DFS server redirects the request to the server that actually hosts that resource. This feature can be powerful, as it hides the distributed nature from the end user. Instead of having to know that Human Resource documents are at [\\server143\hrdocs](#) and sales presentations are at [\\server192\salespres](#), the user can attach to [\\server1\fileshare](#) and locate any folder as a subfolder underneath. The additional power of DFS is the power of replicating the information so that several servers have a copy of the information. Thus, if you need to perform maintenance time on server X, you can take it down and add new drives.

That is how Windows Server DFS is designed to work on the surface. But note a few cautions: As with any product based on replication, you must proceed carefully or risk creating a topology that is unable to complete its replication. Too many sites over slow links will cause the topology to fall apart. DFS works great for LANs, but be sure to test the impact on available bandwidth before designing across WANs. Based on experience, it is a difficult task to develop a rule of thumb stating that you can replicate a DFS topology of x servers, holding y gigabytes over a WAN of z bandwidth. Regardless, DFS-controlled replication can be advantageous over users pulling files across the WAN during peak business hours.

Another area of caution is that SRM products aren't really designed with a DFS topology in mind, but how could they be? The DFS is designed to consolidate file shares, and SRM products, must operate directly at the server. For example, StorageCentral SRM loads a driver that you would need to load on each server in the DFS topology.

Windows Server RSS

Rather than providing additional storage through DFS, perhaps the time has come to create a secondary tier of storage using cheaper (and slower, unfortunately) storage such as tape to ease the burden on the primary storage systems. Enter the world of HSM in Windows Server through the new RSS. This service is based on an application originally developed by Seagate (which is now part of VERITAS). Thus, if the Windows Server version is enough to whet your appetite and you're serious about this functionality, there is a full-blown version available.

 RSS is not included in WS2K3 Standard Edition, you must upgrade to the Enterprise Edition; WS2K3 Datacenter Edition also includes RSS.

HSM makes a lot of sense when you look at how you and others in your company access data. Access is hectic for a few days or weeks, then the files sit just to satisfy the Murphy's Law that says "don't delete that file or else you'll need it." Key to HSM is that it looks no different than ordinary file storage to the end user, except retrieving a file takes a long time. Especially during busy times, HSM may saturate the system and test the patience of the end users. Worst case is a file that is accessed just infrequently enough to be rotated to tape but is then requested by many users.

To set up HSM or RSS in Windows Server, the administrator defines a policy such as the amount of free space to maintain on a volume, which files are candidates for migration to the secondary storage, and the latency period (how long the system should wait before a file is moved to secondary storage). To pull off this magic, Windows Server maintains a database that contains the reference to the storage of the actual file. When the file is accessed, it is copied from tape to disk. RSS depends on the Windows Server Removable Storage Manager to handle all the device library functions, such as mounting and dismounting media.

If you can, give RSS a try and see whether users can put up with the latency of offloading files to tape, or whether you should add more disk-based storage. Whether you decide to add a new storage system because you're performance constrained or storage constrained, let's look at the hardware-selection process and the decision points for the types of physical storage and location of physical storage. Perhaps you're even considering additional servers and storage to build your DFS. We'll look at storage design and selection criteria in the next section.

Supported Storage Systems

At the highest level, there are only a few types of storage architectures that you can deploy on your Windows Server systems: DAS, SAN, NAS, and a new type of storage architecture based on the iSCSI protocol (this architecture is somewhat of a hybrid product and is new to the market). The server-to-storage relationship can be one-to-many, with one server attached to multiple storage devices or arrays, or many-to-one, with one storage network (either SAN or NAS) attached to multiple servers.

The many-to-one model introduces new levels of complexity and issues that must be dealt with, namely that anything that is shared must be managed as such, and conflicts in shared access must be dealt with gracefully. Unlike DAS, both NAS and SAN introduce the possibility of heterogeneous environments in which Windows Server must share storage with other OSs. We'll consider this important aspect as we look at NAS and SAN.

If you're building a server from scratch, you pop in some form of removable media (CD-ROM or diskette), boot to that media, copy files to magnetic media (a hard drive), and once the OS is running, configure other devices such as tape or rewriteable optical media. These are all forms of DAS. By its nature, DAS can be used only by the server to which it's connected, and it's connected by a channel (usually a SCSI cable or an IDE cable) that connects all devices directly to the system bus. As the need for more advanced forms of computer architectures, such as clustering, drove the need for shared access to storage devices, the shared architecture of SAN arose.

Networked Storage

The SAN is often depicted as a cloud (similar to a network cloud through which clients gain access to servers) that lets any number of hosts (servers) be attached to any number of storage devices (including disk, tape, or other media). Alternatively, NAS is a network cloud that exists between the application servers and the storage devices. Each NAS system is known or accessed by a name or IP address, just as you would address a server. Figure 4.8 illustrates the DAS model versus SAN and NAS models.

There are a few emerging hybrid technologies on the market. Most of these products provide a protocol bridge between SCSI (used natively in DAS), fibre channel (used in the SAN), and IP (used for NAS). The protocol bridge allows multiple hosts access to multiple storage subsystems, but the storage systems typically remain isolated from each other—the storage systems retain the unique characteristics of DAN, SAN, or NAS.

An upcoming technology is *storage virtualization*, the ability to treat all storage subsystems as one big pool and allocate useable space much more dynamically with less concern for individual cabinet or subsystem boundaries or storage capacities. I'll provide more information about this newly emerging technology in future chapters. For now, the virtualization capacity exists primarily on a single storage system or on a single storage network.

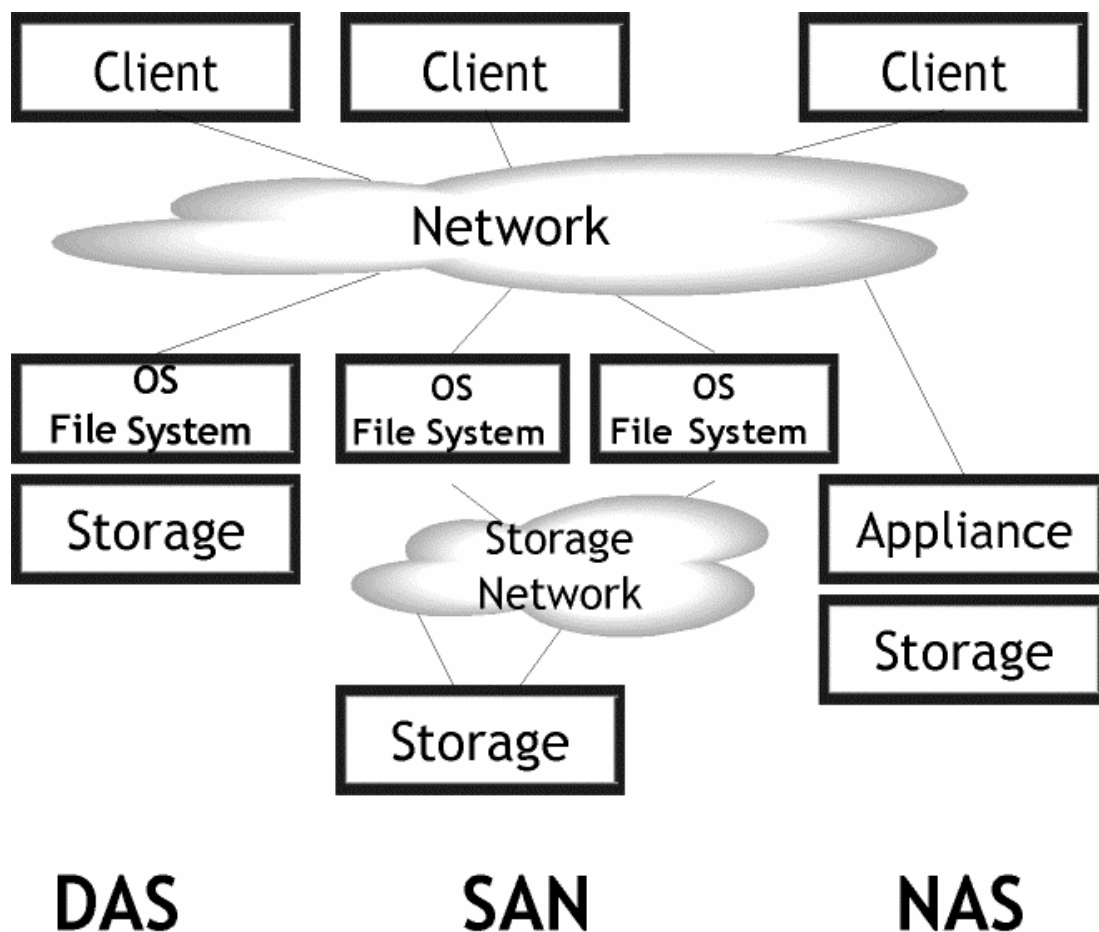


Figure 4.8: DAS model versus virtualized storage model.

These islands of information soon became networked, sometimes as a peer-to-peer network, which continued the same problems of isolated pockets of information. Centralized computing through the client/server model placed the information in a shared access repository, often only isolated by the reach of the network.

Enter the Internet era, which removed even more barriers, by extending the network and flattening the standards for information sharing (through the Web browser and XML), reaching a broader variety of mobile devices, such as the Internet-capable Windows CE PDA that I'm using to type this book in-flight. Much of the growth in Internet-enabling traditional brick-and-mortar companies has been in making data repositories accessible through the browser and allowing business-to-business communication or data transfer through a common standard language.

Does this same removal of boundaries apply to storage? Most definitely—any barrier that says “It's not that way, it's over there” must be removed or at least virtualized so that the end user connects to information at the highest level and doesn't have to make a decision about which file format to retrieve from which application on which server on which network.

Virtualization

Virtualization is a key concept in storage and computer systems in general. From my viewpoint in the computer industry, the progress of the past 20 years has been about removing barriers to information flow and access. We don't need to go into a lengthy discussion about the demise of the mainframe, but keep in mind that some of the same concepts will apply to removing storage barriers. The benefit of PCs is that they allow individual workers the ability to perform computing on their own schedule. You can play what-if on your spreadsheet or compose and revise a memo (before printing it out and sending it through the postal mail) without having to schedule mainframe time or otherwise interfere in IT operations.

Virtualization also plays a key role in fault tolerance and performance, as can be seen from load-balanced Web server-caching farms. The user connects to a virtualized namespace served by any one of a number of Web hosts, which may even contain a virtualized view of the data repository. The same concept can be found in storage, as an application server connects to a pool of fault-tolerant storage, accessing its allocated storage as opposed to its installed devices. On the OS side, you need an OS that is aware of this virtualization or at least tolerates it more than NT does. (NT tends to want to own any device it can see.)

So storage can be direct attached, removable (on a SAN or NAS), or some hybrid mixture. A differentiating characteristic is cost. In the area of cost, I'm not a big fan of SANs—they're expensive beasts. However, if you must create shared storage or use advanced volume-cloning or replication features, SANs are your best choice.

NAS has been coming on strong by providing relatively less expensive storage that can also provide some data-protection features. Although NAS is solving one problem by giving you a massive amount of storage space, it's creating another problem in that you must somehow manage that storage. Think of all the storage-related applications or functions that you can add to your NT and Windows Server servers—backup, antivirus, performance monitoring, utilization monitoring, quota management, file recovery, and so on—and consider how you'll make sure that you have the same level of functionality on your NAS devices. A final note about NAS: NAS can't be used with some applications, such as Exchange Server. Thus, nothing can beat DAS for cost-effective raw performance. By the time this book is complete, next-generation RAID controllers will hit the market that can outperform the high-end SAN controllers provided by those huge data-storage systems targeted at the Intel market and beyond.

Storage Solutions

The storage market is characterized by typically having a few dominant players in each category or type of device or product, and many categories of products are necessary to put together a complete solution. How does this market setup apply to your Windows Server storage deployment?

You'll most likely be using products from a wide variety of vendors to put together an end-to-end solution (from hardware to software). One bit of advice about the difficult process of selecting storage devices and storage-management products: No doubt, you'll put a lot of work into research, through reading or talking to others at technology events. You'll probably come into contact with what I call technology bigots. Personally, I try to keep an open mind and keep an eye out for new solutions. Technology bigots will attempt to sway you or cause you to question the value of the solutions that you put together. They'll claim that your product vendor is no good and that the one they use is the right one. Some people just feel the need to justify the expense of their solution. Many times what is actually taking place is comparing two completely different solutions (like comparing a 4 × 4 truck with a two-seater roadster). Is one solution better than the other? Yes, for approaching different problems or needs. You might find that using products and solutions from a vendor that is familiar to you and well-tested in your Windows Server Intel architecture is more appropriate than being swayed by some massive storage behemoth lumbering down from the mainframe world.

The same goes for the storage-resource management solutions—I like to focus on the ones that have a solid background in NT and Windows Server, and solve my pains as a Windows Server administrator. It's OK if the product can talk to 16 flavors of heterogeneous environments, but it had better be the best choice in the environment that I care about—Windows.

Storage Service Provider

Another form of storage architecture that you might consider or already find in place within an organization is the use of a Storage Service Provider (SSP) for outsourcing storage needs. The appeal of the SSP is that you can purchase (or lease) your storage on a monthly basis and avoid the associated headaches of storage expansion: space, power, cooling, and so on. This solution might be appealing for the start-up company that faces growing storage needs but isn't sure that it can survive through the next few years. I'm just mentioning this option as a possibility, although Windows Server makes hosting your own storage easy enough. Perhaps even easy enough that you can offer storage as an outsourced utility that you resell to others, assuming that you have the bandwidth to your facilities. The most likely benefit that an SSP can provide is keeping a secondary data set at off-site storage for disaster recovery protection.

Storage takes a lot of bandwidth—we're not talking about network transmissions, we're talking about moving huge amounts of data to and from channel devices. Let's take a look at these devices.

Storage Devices

In deploying your storage, you'll need to choose amongst myriad storage devices. We'll look at the characteristics of each type to help you decide. No doubt you'll be constrained by financial resources and must make some tradeoffs—should you spend more to get that 15,000rpm drive or stick with the 10,000rpm drive and get more drive space? Meanwhile you must protect your investment from obsolescence before you have achieved satisfactory return.

Most likely you can relate to the experience of explaining to a non-technical person how a computer works. Remember the frustration of dealing with the fact that he or she could not keep the concept of memory (RAM) distinct from storage (the hard drive). You were forced to clarify storage as being distinct from memory in the following areas: used for holding larger amounts of information for longer terms, sacrificing performance for a more cost-effective mechanism to increase capacity. At times the boundaries may blur, with memory technologies being used for storage, but the distinctions still hold true. Earlier computer systems considered memory as primary storage and other mediums secondary storage, such as magnetic tape or even punch cards. You may be familiar with the concept of a RAMDRIVE, which emulates the properties of a storage device by presenting memory as a drive volume—with one exception: information is lost when power is no longer supplied. This exception is where we'll draw the line between storage and memory; a storage device must not be quite so volatile. The characteristics of the RAMDRIVE prevent any portability between systems, which is important functionality of many storage devices.

From a practical standpoint, storage isn't just about capacity and speed. Storage devices' so-called abilities are extremely important: reliability, availability, scalability, and manageability. We'll take a look at applications that can be added in each of these areas as well as define the necessary processes to achieve these abilities.

As Figure 4.9 shows, performance typically comes at a greater cost per megabyte.

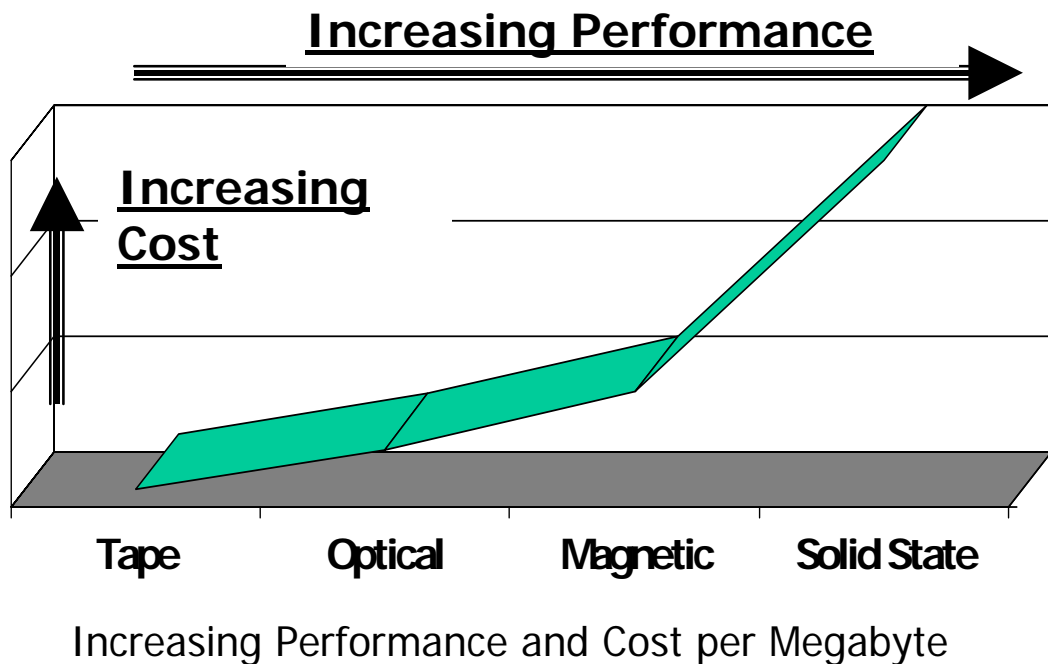


Figure 4.9: Relative positioning of storage devices.

Performance can be measured in many ways. The first is in the data-access method; whether it is random access, at any point on the device in near real-time, or linear access, which can take a substantial amount of time to reach the end of the media. After the initial data point is located, performance is often a measure of access speed: retrieving the bits and sustained throughput and moving the entire chunk of information to the requestor. At some point, we must make a purchasing decision: which type of storage provides sufficient performance at a reasonable cost. Most likely, the answer will be magnetic disk drives.

Disk Drives

As you plan your storage, you'll need to select storage devices, so I'll provide a quick overview of the types of storage devices supported in Windows Server. The main types of storage devices are hard drives, solid state drives, tape, and optical drives and media. The most common type of storage technology that we'll work with is magnetic disk drives. Most of the time when I talk about storage devices, I'll be talking about hard drives. Hard drives, hard disks, and fixed disks are all synonymous for storage that sits near the middle of the chart in Figure 4.9.

Storage Density

For a while, hard drive vendors had reached a ceiling and squeezing more storage into the existing form factor of hard drives was becoming increasingly difficult. The problem is that the magnetic particles used to store the data on the spindle platters can be only so close before they start interfering with each other. There are some workarounds to this problem such as increasing density, for example, as the IBM pixie dust breakthrough claims to quadruple disk drive density.

From your knowledge of the storage market, you know that hard drive density has been steadily increasing. A hard disk drive is typically composed of multiple platters, all of which are accessed as a *spindle*, the total formatted hard drive capacity. Single platter density has been increasing, recently reaching 40GB, and is expected to double to 80GB over the next year. The next doubling, to 160GB, will take place the following year, but will be of less interest until a barrier at 137GB is broken. The 137GB barrier exists because current drive technologies use a 28-bit architecture for accessing data and can't handle greater capacities. Maxtor and other industry players such as Microsoft are working on 48-bit logical block addressing in the disk interface that will allow as many as 144 petabytes of data (once the 32-bit address space of most OSs grows beyond the 2.2 terabyte limit).

Since the greatest storage capacity is achieved with multiple platters, you may be asking why hard drive manufacturers would ever build a single platter drive. If you actually had the choice of buying two drives of 40GB capacity for the same price and found out that one was multiple platters and the other was single platter, why would you prefer one to the other? The reason that you might pick the single platter drive is that it should achieve greater levels of reliability because the single platter means fewer mechanical components in the drive. It might even be a smaller form factor and cost less. But the multiple platter drive most likely will outperform it, for reasons that we'll soon see.

SCSI versus IDE

When designing high-end storage systems, SCSI hard drives are preferred over IDE drives for many reasons. IDE drives are more common in personal desktop systems, but are definitely available in some NAS. SCSI drives have always been faster than the fastest IDE drives and allow more devices in larger disk arrays. But a newer type of IDE drive, referred to as a Serial ATA (SATA) drive has entered the range of SCSI performance. Boasting speeds of as fast as 150MB/sec (in short durations, rarely sustained) the SATA drives are also more highly instrumented, like SCSI drives, meaning that they use self-monitoring to allow alerting on predicted drive failures. This is important, as the SATA drives do not have the hardware reliability of the SCSI drives and must be monitored.

Until recently, only SCSI drives have been used in RAID systems, but IDE RAID is certainly available and could provide fault-tolerant drive protection for a small workgroup server. However, IDE can add only two devices per channel (with most IDE controllers offering only two channels), and SCSI definitely has the advantage of supporting chains of 8 or 16 devices. Table 4.4 compares IDE and SCSI standards as well as Fibre Channel, which is essential for SAN. The variety of choices can be confusing; fortunately, newer designs such as ULTRA3 for hard drives are backward-compatible even though the cabling usually isn't.

Channel Name or Standard	Transfer Speed (Maximum)	Comments
UDMA 33 (IDE)	33MB/sec	Technical specification for the channel; however, rarely achieved in any duration especially with the IDE-device limitation of two per channel.
UDMA 66 (IDE)	66MB/sec	Same as above.
SCSI or SCSI-1	Asynchronous data transfer rates of 1.5MB/sec and synchronous transfer rates of 5MB/sec	As many as seven devices per adapter; narrow (8-bit), single-ended cabling as long as 6 meters; SCSI bus clock rate of 5MHz.
SCSI-2	A standard, the speeds are defined by the implementations below.	The standard that defines the differential interface, allowing as long as 25-meter cable length and single-ended SCSI cable length as long as 3 meters. Adds 16-bit and 32-bit wide data bus, doubles data throughput by doubling the clock rate, and a smaller 50-pin, high-density connector. Includes other improvements in reliability through synchronous negotiation and parity checking.
FAST SCSI or FAST SCSI-2	10MB/sec	Narrow (8-bit) at 10MHz; doubles the SCSI bus clock rate of 5MHz.
WIDE SCSI	10MB/sec	16-bit at 5MHz but requires two cables (A+B). The wide designation indicates a 16-bit data path.
FAST WIDE SCSI	20MB/sec	Transfers data over a 16-bit wide SCSI bus at 10MHz.
SCSI 3 and SPI	A standard, the speeds are defined by the implementations below.	Establishes the SCSI Parallel Interface (SPI) using a 68-pin, high-density connector for 16-bit wide SCSI.

Channel Name or Standard	Transfer Speed (Maximum)	Comments
ULTRA SCSI	20MB/sec	Also called Fast-20. Doubles the FAST SCSI throughput for 8-bit and has the same cable lengths as WIDE ULTRA SCSI.
WIDE ULTRA SCSI	40MB/sec	Doubles the FAST WIDE SCSI throughput to 40MB/sec for 16-bit. Maximum single-ended SCSI cable length of 1.5 meters for five to eight addresses (devices) and 3 meters for four or fewer addresses. Maximum differential (HVD) SCSI cable length of 25 meters.
ULTRA 2 and SPI-2	A standard, the speeds are defined by the implementations below.	A second SCSI Parallel Interface (SPI-2) that doubles the bus speed to ULTRA 2 (Fast-40) SCSI throughput. Uses a new electrical interface known as Low Voltage Differential (LVD) as opposed to the older TTL-based differential SCSI (now called High Voltage Differential or HVD). SPI-2 adds two new connectors, the 80-pin Single Connector Attachment (SCA-2, which includes the 16-bit SCSI signals and power for the peripheral) and the Very High Density Cable Interconnect (VHDCI) connector, a small 68-pin wide SCSI connector allowing for as many as four connectors on a controller back plate.
WIDE ULTRA2 SCSI	80MB/sec	16-bit data path with ULTRA 2 SCSI SPI-2, effectively doubling the bandwidth. Maximum cable length of 12 meters.
Fibre Channel	100MB/sec to 2Gb	100MB/second is the single-loop speed. Fibre Channel is capable of combining multiple paths in a SAN to boost channel bandwidth several times. Encapsulates channel transmissions for SCSI devices on both ends. Maximum cable length of 10km to 100km (depending on the cable type and wavelength), far exceeding the native SCSI capabilities without SCSI extenders. Can theoretically support 126 devices in a loop and more in a switched environment.
ULTRA 3 and SPI-3	A standard, the speeds are defined by the implementations below.	SPI-3 doubles the SCSI bus speed to Ultra 3 (also known as Ultra160 and Fast-80). Requires a 16-bit minimum bus width and uses Double Transition (DT) clocking (on both the rising and falling edges). Also includes Domain Validation (a write verification test at full data rate, which will fall back to the next lower speed if not successful) and a 32-bit cyclic redundancy check (CRC) for better data integrity. U160/m is a limited set of SPI-3 including these three improvements: DT clocking, parts of Domain Validation and CRC. SPI-3 also defines Quick Arbitration and Select (QAS), and Information Units (Packetization). Maximum cable length of 12 meters or 25 meters for point-to-point.
ULTRA160 SCSI	160MB/sec	Also known as Ultra 3 SCSI. Based on Fast-80, capable of 160MB/sec as long as 12 meters.
ULTRA320 SCSI	320MB/sec	Also known as Ultra4. At this point, the next step in the development of SCSI standards, not a production technology yet.

Table 4.4: Channel standards and resulting speed.

Now let's turn our attention to designing a storage system with fault tolerance to protect our data against catastrophic loss (or even ordinary, everyday events).

RAID

RAID was designed to eliminate individual disk spindles as a single point of failure. Hardware RAID controllers, sometimes with vendor-specific designs or feature sets, can implement RAID. RAID can also be implemented in the OS itself, through software RAID in Windows Server or an application. Table 4.5 shows the different RAID levels.

Unlike the others, RAID0 is merely a multi-spindle technique and provides no fault tolerance. In fact, it increases the probability of data loss: as more spindles are added to the stripe set the chance of one failing will increase. The other RAID levels offer distinct advantages in certain situations. The same approach for fault tolerance or performance striping can also apply to other storage devices, such as Redundant Array of Independent Tapes (RAIT).

RAID Level	Implementation	Percentage of Useable Storage	Explanation	Comments
RAID0	Stripe set	100 percent	Data blocks written across several spindles	No fault tolerance, but best performance
RAID1	Mirror set	50 percent	Same data blocks written to two spindles	Typically a two-spindle implementation; for more disks, see RAID0+1
RAID2	Hamming error correction	n/a	Provides error correction for drives that don't have built-in error detection	Rarely implemented, as SCSI drives provide error detection
RAID3	Byte-level striping with parity spindle	n-1*	Similar to RAID4 dedicated parity spindle, except byte-level striping	Not widely implemented due to intense activity and increased likelihood of failure of parity spindle (although read performance similar to RAID0)
RAID4	Known as Data Guarding; block-level striping with parity spindle	n-1*	Similar to RAID5 block-level striping, except dedicated parity spindle	Not widely implemented due to intense activity and increased likelihood of failure of parity spindle (although read performance similar to RAID0)
RAID 5	Known as Distributed Data Guarding; stripe set with parity	n-1*	Data blocks and information necessary to rebuild any missing blocks if a single spindle fails; written across several spindles	Write-performance penalty, as all disks must be read, parity recalculated, and all disks written to, to update information on disk; some read performance impact as parity information must be read-over or skipped

RAID Level	Implementation	Percentage of Useable Storage	Explanation	Comments
RAID0+1	Mirrored stripe sets	50 percent	Stripe sets mirrored to another stripe set with the same number of spindles	Can be considered the same level of fault tolerance as RAID10, depending on whether it can survive loss of two disks in the same stripe set
RAID6	Multiple parity stripes	Less than n-1	Similar to RAID5 but a second parity calculation is striped across all drives, adding more fault tolerance	Rarely implemented
RAID10	Striped mirror sets	50 percent	Mirror pairs combined into a stripe set	Can be considered the same level of fault tolerance as RAID0+1, depending on whether it can survive loss of two disks in the same stripe set; typically a vendor-specific design
RAID3/5 or RAID53	Hybrid approach	n-1*	Vendor-specific hybrid approach	Designed to remove the parity update penalty of RAID5 and soften the impact to the RAID3 drive; often a RAID-controller specific implementation (not really RAID3, but RAID3-like)
Other	Vendor specific	n-2*	A variety of other techniques such as Advanced Data Guarding (ADG), which writes parity information to more than one spindle	Designed to provide more fault tolerance than other RAID without sacrificing performance
*n-1 useable storage for RAID means that one spindle in a set of n is given up for parity information and is not available for data storage.				

Table 4.5: RAID implementations and analysis.

RAID Controllers

As I previously mentioned, RAID support in Windows Server is built-in through software RAID and can also be purchased from third-party vendors as software or hardware (RAID SCSI controllers). As is often the case with other features and functionality in the OS, Windows Server's software RAID can provide only limited functionality in certain circumstances and will fall short for many applications. The built-in software RAID can be used to mirror (RAID1), stripe (RAID0), or parity stripe (RAID5) disks.

Hardware-based RAID justifies its cost by providing high performance and manageability. The high performance is achieved by adding high-speed cache and a dedicated processor, essentially a dedicated server on a card. Replacement of failed disks in RAID sets and recovery of disks is usually much easier with a RAID controller, as long as the following caution is observed.

🔴 Highest performance is achieved on a RAID controller by enabling write-back caching, which acknowledges to the OS that the data is safe on disk even though the data is really in the controller cache waiting for the disk array to accept the data. In the event of a sudden loss of power, the data would be lost unless the controller or the server has a battery (Uninterruptible Power Supply—UPS). Some RAID controllers have a battery-backed cache onboard, which is a must if enabling write-back caching, but the UPS is a must to protect the server, especially if the hard drives use caching.

Hardware RAID purists can be rather snobbish at times, turning their nose up at software RAID, but there is a definite need and benefit for using software RAID. Hardware RAID is the way to go for high performance and a wider variety of migration and recovery options. But software RAID can be useful when riding on top of a hardware-based RAID controller. For example, software RAID enables you to create larger disk volumes than are possible using the hardware RAID controller itself. Recovery of software RAID in NT can be tenuous, as information is stored in the system registry. But in Windows Server, the information is stored on the disks involved in the form of metadata, which means that the storage should be recoverable even in the event of total system loss.

Performance Design

Now that we've looked at fault tolerance, let's take a look at performance considerations. Because the costs and capacities of storage have been dropping rapidly, we're able to store greater amounts of information. But the speed of moving the information from the hard disk into the system memory hasn't grown at the same rate. So what you end up with is a relative loss in serving up information—much more information moving slightly faster means that storage can become the bottleneck in modern information systems.

There are two broad categories of devices that you're probably already familiar with: random access and linear access. Hard drives are random access and can quickly find data located anywhere on the drive. Tape drives are linear access and must move a portion of the tape, which varies depending on how far from the beginning the data is. Even hard drives have a delay inherent when seeking the location of information, though it is nowhere near as much as for a tape drive.

If a request for data is made and the head is not in the right position, it must wait for the platter to complete its rotation before it can begin serving the data. Much emphasis has been placed on rotational speed, as SCSI drives have increased from the 5400rpm to 7200rpm range to a 10,000rpm to 15,000rpm range. Rotational speed typically has a large impact on performance, but the point to understand is that rotational speed is only a single factor affecting performance, and it may not be the limiting factor.

Performance of a storage subsystem is measured in several variables. First is the access speed, or how long it takes for the storage device to be ready to read or write a requested piece of data after an access command has been issued. Once the data is found, there is a time period, or latency, that it takes to move the bits from the storage medium to the channel. Once the data is in the channel, another measure of performance is how fast the device can continue to deliver the data—the measure of sustained throughput or data-transfer rate, often referred to as bandwidth. Oddly enough, this measurement is the performance criterion most people are familiar with and is often a factor determining purchase, even though it may not be the most important criteria for system performance. Quite often, applications are characterized as being either bandwidth intensive (the most common example being streaming media such as video) or input/output (I/O) intensive (a common example being a transactional database).

The total access time of a storage subsystem is the sum of all these factors: seek time, latency, rotational speed, bandwidth, I/O capacity, and controller overhead. For example, an array of several drives may be replaced with drives offering a faster rotational speed, but no performance gains may be achieved if the controller is the limiting factor or if the number of drives in the array can't sustain the I/O requirements. I'll illustrate this idea with a performance design example in a moment.

A typical SCSI hard drive listing may list the following attributes: 36GB, U160W, 6.7ms, 68Pins, 7200rpm, 4MB cache. At this point, we should know enough to decipher this hard drive as being an ULTRA3 SCSI hard drive with a 6.7 millisecond seek time, and a rotational speed of 7200rpm. The connector uses 68 pins, introduced in Table 4.4 under the ULTRA2 VHDCI connector. The 4MB cache helps to speed disk access as the high-speed chips reduce the occurrence of physical disk access. Most likely you'll be asking "Is this disk the right disk for the job?" as you compare it with higher-priced offerings for faster rotational speeds, lower seek times, and greater capacities.

As you can see, there are quite a few variables to consider when designing the optimal storage system. The following list provides questions to ask in determining the optimal storage system:

- Application characteristics: Is it bandwidth intensive or I/O intensive?
- What is the mix of read/write ratio?
- Does it perform sequential or random I/O patterns?
- Does the application have several components that perform with different characteristics?

Through most of this book, we look at the primary use of Windows Server storage as file-sharing services, but we also look at managing application servers as a storage resource. To illustrate an application server with a mix of performance characteristics, in the next section, we'll look at Exchange Server 2003.


Performance Design: Exchange Server Example

Let's look at a Windows Server application for a moment because it helps to illustrate many of the different types of performance characteristics. For example, an application such as Exchange Server 2003 acts as several types of services or applications and performs a mix of several patterns. As a result, the optimal storage design for an Exchange Server, from a performance standpoint, is to build separate storage volumes for each type of disk access. Let me emphasize this point: the biggest bang for the buck comes not from tweaking disk array parameters but from designing separate storage based on disk access patterns.

To illustrate, Exchange Server is a transactional database, which means that every incoming transaction must be committed to a log file in case there is an interruption of service (power loss or server crash). Each transaction is a messaging action, such as when you open an email, or when you type a new email message and save it to a particular folder. Transactions are the smallest unit of work that the Exchange Server needs to keep track of (to ensure that they're not lost or unnecessarily repeated in the case of server crash and recovery). The write to the log file is sequential, and once it is committed to the log file, it can also be committed to the database volume. No reads need to be made from the log file as long as the current transaction is in memory, which it will be, unless the server had a hard crash and needs to replay the transactions from the log files to determine where it left off. This circumstance, the reading of log volumes, is unusual enough that you would not design the log drives for it, and instead design them for continuous sequential writes.

The database part of a transactional database system is responsible for writing these committed transactions to the database files. Quite often, this functionality extends or grows the size of the database file, although this growth isn't always the case, as it could write the transactions to any available white space in the database file (space that has been freed up by deleting items). While the database volume is writing these transactions, such as email coming in to your Inbox, you're also reading other email or checking your calendar, which means that the database must be read. So you have a random mix of read/write on the database files, usually in the range of 50/50 that is quite a different pattern than on the log files. Thus, for highest performance, the log files are placed on a separate volume than the database volume.

In Exchange Server 5.5, this process was much simpler than it is in Exchange Server 2003, which provides the ability to create multiple storage groups within the same server information store process—each storage group having its own set of log files. So the tough question became “Do I place each set of log files on its own disk array or should they share the same log disk?” If they share the same log disk, the disk access is no longer pure sequential writes, and the disk heads must move between multiple locations on the disk, much as they would on a heavily fragmented drive.

 Each log file array would most likely be a mirrored set of two spindles; thus, creating two logical volumes on the same array hardly makes sense. It might help from an administrative or logical view, but it would do nothing from a physical performance standpoint—you would still have the loss of pure sequential writes.

Exchange 2000 Server introduced another database file format to the information store, known as a streaming media file (it has a .STM file extension). The purpose of this file is to allow direct reads and writes in native Internet protocols—such as SMTP, POP3, and IMAP—for much faster disk access and higher bandwidth streaming. This functionality can have an impact on your storage design, as it gives you the possibility of separating the .STM files from the .EDB files for higher performance. The size of the database pages is also enlarged from 4Kb for the native Exchange EDB to 32Kb for the .STM file, so larger writes can be performed. This factor can be taken into account when selecting the size of the allocation unit when formatting the disk. So a possibility is to design an Exchange Server with multiple disk arrays, for each set of transaction logs associated with each storage group (for the .EDB database file and possibly even the .STM database file).

Another reason to use Exchange Server as an example for designing storage subsystems is that it also performs another type of disk access: an Exchange Server can exchange messages through SMTP queues, either as part of inter-server routing or as part of SMTP routing to the Internet. The SMTP queue disk access pattern is a constant stream of writing messages in and reading messages out. Messages must be committed to disk, to prevent data loss, before they're forwarded or delivered to another SMTP server.

Designing for Windows Server file and print services may actually be a bit less precise, as the read/write mix and patterns may shift over time, making it more difficult to define a specific storage volume. If there is a situation in which you can separate the sequential, read-intensive disk access from the random write-intensive pattern, this separation would be beneficial, just as in the example of Exchange Server.

Let's apply what you have read so far to designing a streaming media server that will have online training videos (such as Windows Media or Real Media files) uploaded to it for trainees to watch. Because the files change infrequently and read performance is paramount, you would place the video files on a RAID5 volume for best read performance and fault tolerance. If fault tolerance isn't important (if this is a replica or mirror of another server and the data drives can be easily rebuilt) then RAID0 would provide excellent performance.

Summary

In this chapter, we covered two areas: structuring the SRM project and Windows Server functionality. We looked at various ways of creating additional storage capacity—from linking systems together through Windows Server features such as DFS and volume mount points to adding storage systems. I briefly covered storage hardware to make you aware of some of the design features to look for and some of the tradeoffs between performance and capacity. Also important is the choice of storage applications and storage location alternatives, as this selection affects your choice of SRM product and process. The outcome of your storage analysis and planning is both the storage-management decision points—such as whether to add more storage to the management problem or to make better use of your existing storage capacity—and the development of an organizational storage policy.

We exhausted the capabilities of the core Windows Server functionality and SRM features, including quota management, before turning to a more comprehensive solution. I detailed the product functionality that assists you in reaching SRM goals through setting disk quotas: eliminating duplicate files, eliminating unused files (aged and orphans), eliminating wasted space, and reducing excess consumption.

In the next chapter, we will look at using storage management tools to make better use of either your existing storage or your newly deployed storage. We'll cover testing the SRM solution, getting feedback, and assessing the effectiveness of the implemented solution. I will give you a heads up about problems that you should anticipate, and assist you in creating a communication and education plan to smooth the impact of your project. In addition, I will give you several deployment templates that can be quite valuable in guiding your pilot phase of the SRM deployment.