

Get a Head Start on NT 5.0 with Dfs

Douglass Toombs

(Reprinted from WindowsItPro Magazine)

It's 3 a.m., and my Washington, D.C. server just crashed. A few thousand important Word documents are on that server. But I'm not worried. The reason I'm not worried is that I protected all the critical shares on the server with Microsoft's Distributed File System (Dfs) technology. The Los Angeles, California server will service all user requests from its replica of the Washington, D.C. data, without the users even realizing it.

Although this scenario is a figment of my imagination, the Dfs technology is not. This technology is currently available for Windows NT 4.0 and will soon be available as part of NT 5.0. Dfs will revolutionize the way you distribute files and shares across an enterprise network. By taking advantage of Dfs in your network, you can increase your enterprisewide fault tolerance tremendously.

What Is Dfs?

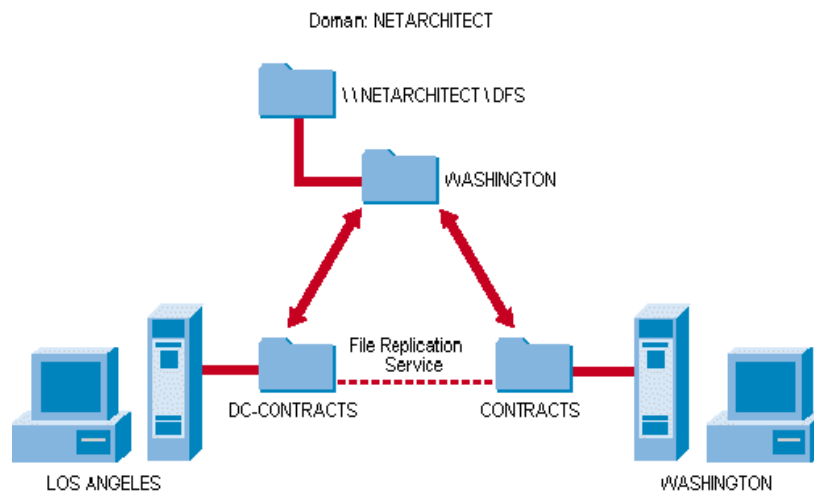
According to Microsoft's Dfs documentation, Dfs implements one name space (or directory tree) for disparate file system resources in an enterprise. You can think of Dfs as a tool that creates a network share that is full of other network shares, either on the same server or on other servers. By publishing shares in a Dfs directory tree, you can logically organize network resources that are physically in different parts of the world. If you organize the resources in one hierarchy, your users can navigate through these resources as easily as they can click through directories.

In addition to centralizing important shares and directories into one hierarchy, Dfs provides fault tolerance and intelligent load balancing by maintaining replicas of essential data on multiple enterprise servers. If one share can't service a user's request, another one simply picks up the slack without the user knowing it.

How Do You Implement Dfs?

Implementing Dfs in a network is easy. Because Dfs isn't a new file system, you don't have to reconfigure existing systems to implement it. In addition, Dfs doesn't affect the NT standard file system's processes. For example, Dfs doesn't alter the NT's permission verification process. If a user doesn't have the rights to access a particular share or file, navigating to it through Dfs will not make a difference. The user still won't be able to access it.

All you need to do to implement Dfs is follow three steps: Create a root directory, add shares, and replicate important data. For example, suppose I want to set up a Dfs tree for the domain in my opening scenario. As Figure 1 shows, domain NETARCHITECT has two servers: WASHINGTON and LOSANGELES. Here's all I have to do to implement Dfs in this domain.

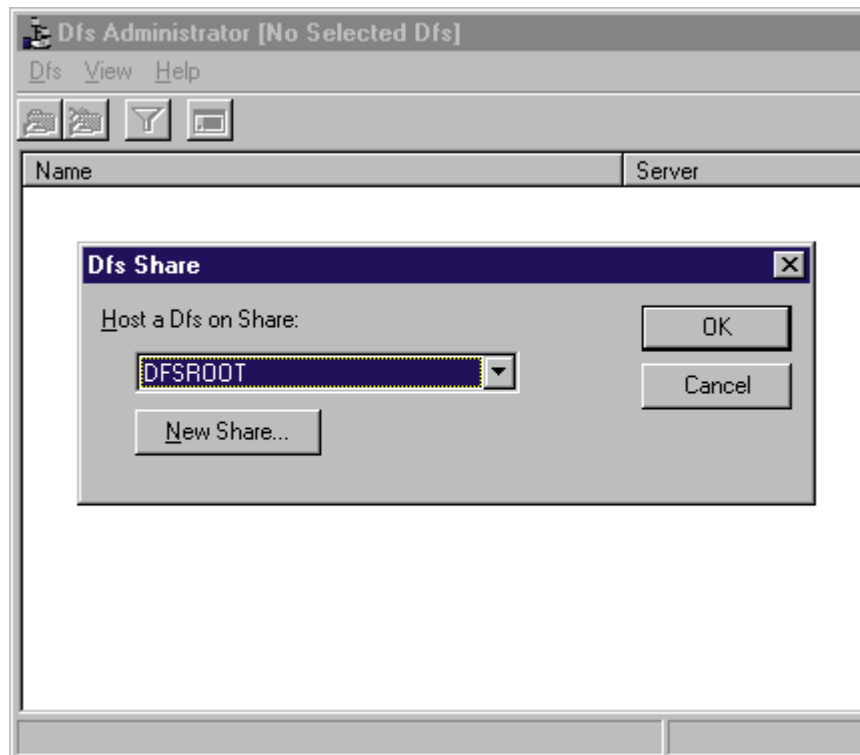
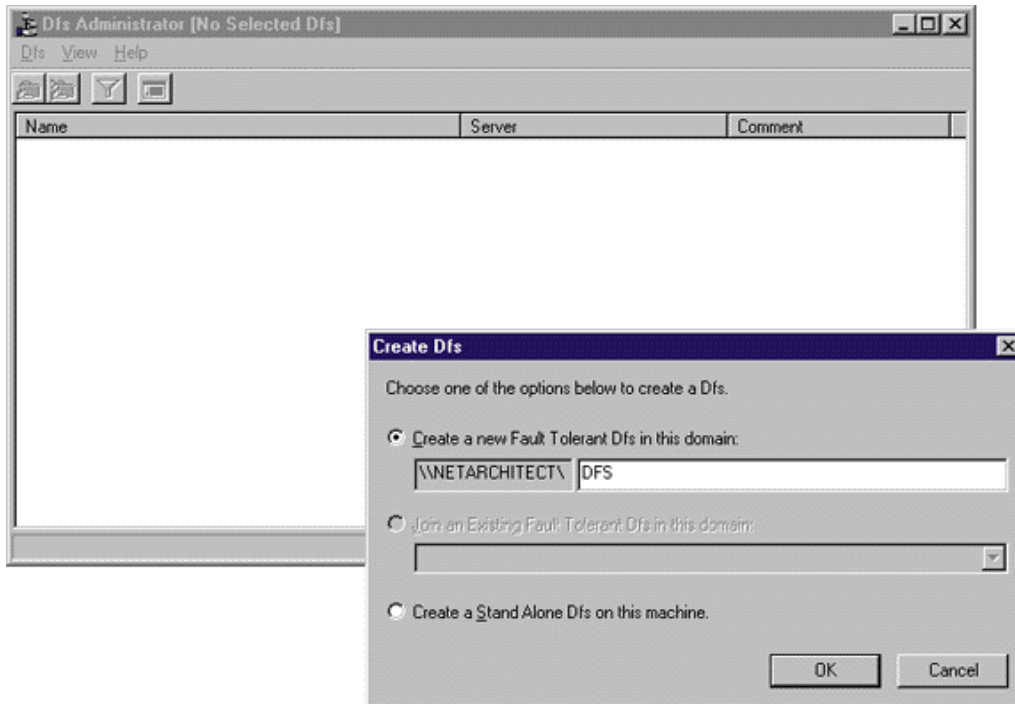


Get a Head Start on NT 5.0 with Dfs

Douglass Toombs

(Reprinted from WindowsItPro Magazine)

STEP 1: Create a root directory. Because every directory tree must have a starting point, I first create a root directory called DFSROOT on the WASHINGTON server and share it under the same name. Next, I use the Dfs Administrator utility to define a new domainwide Dfs tree, as Screen 1 shows. Then, I simply select the share to publish as the root, as Screen 2 shows.



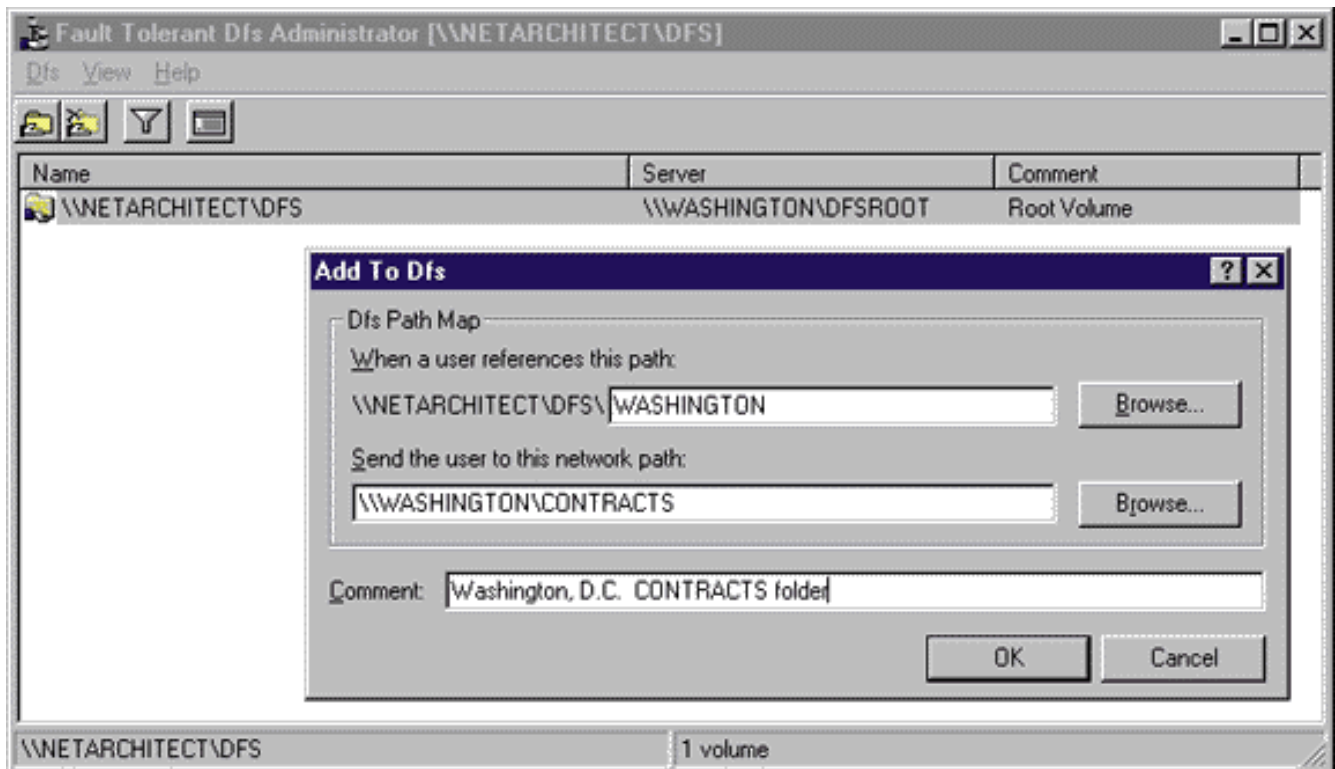
Get a Head Start on NT 5.0 with Dfs

Douglass Toombs

(Reprinted from WindowsItPro Magazine)

I now have an empty root directory for a Dfs tree. This root directory isn't very interesting, but how I use Dfs to get to this directory is. I must use a universal naming convention (UNC) path that references the name of my domain instead of a specific machine. Although the true root of the Dfs tree is at \\WASHINGTON\DFSROOT, the way I map to it through Dfs is \\NETARCHITECT\DFS.

STEP 2: Add shares. Now that I have a root directory, I need to add shares to it and point those shares to resources across my network by editing the properties of the Dfs root. In Screen 3, I'm adding the \\WASHINGTON\CONTRACTS share as a subdirectory called WASHINGTON in my Dfs tree. The Dfs path to access this share is \\NETARCHITECT\DFS\WASHINGTON.



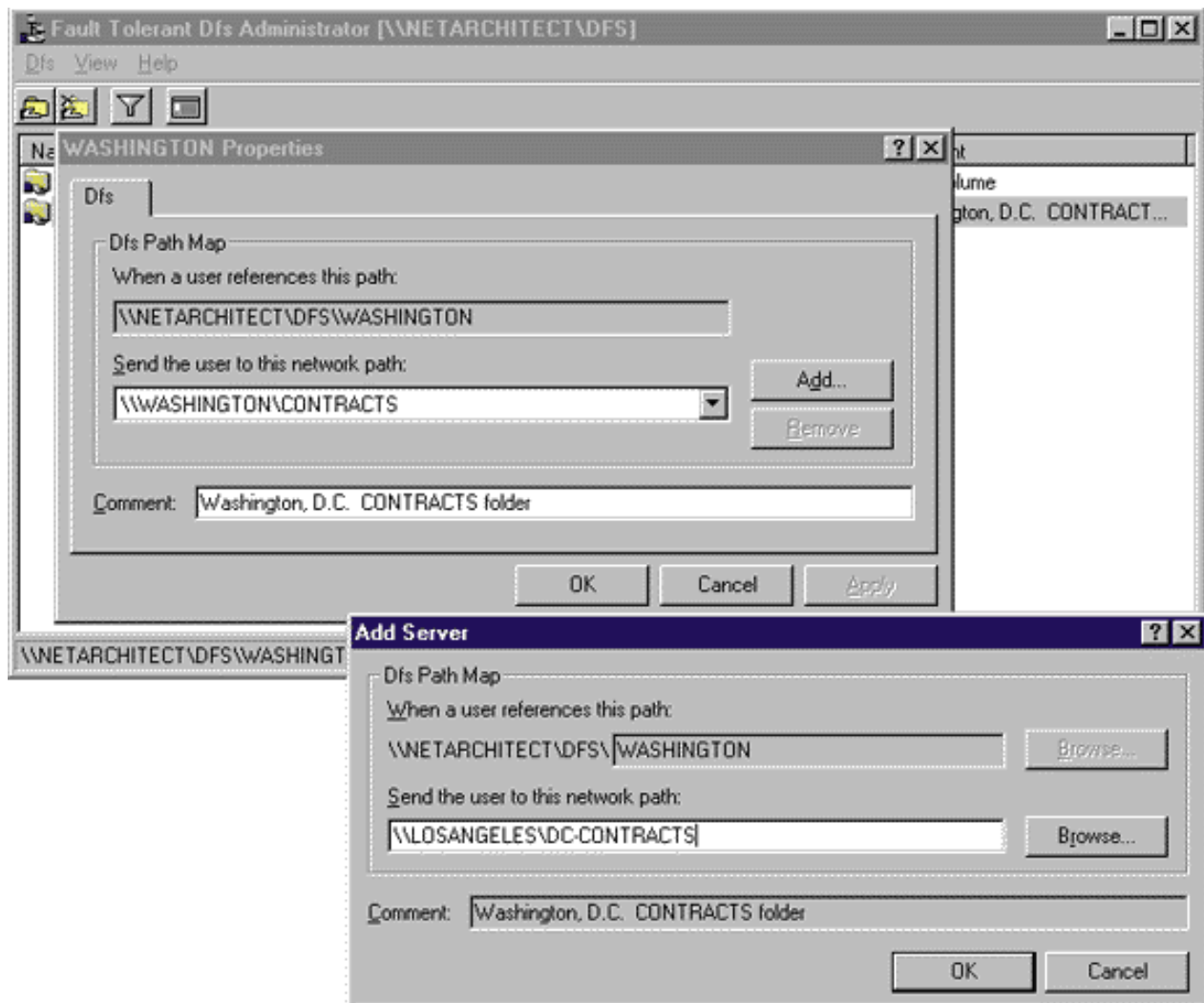
When I add shares to the Dfs tree, I don't have to keep the original share name. I can choose any name I want. This flexibility is particularly useful for resolving enterprisewide naming convention issues.

STEP 3: Create replica sets of important data. Because the CONTRACTS folder is important in my network, I create a replica set for that data. I place a duplicate of the entire directory on the LOSANGELES server by copying the data into the share at \\LOSANGELES\DC-CONTRACTS. Then I add alternative location information for the existing Dfs entry by editing the properties of the \\NETARCHITECT\DFS\WASHINGTON share. As Screen 4 shows, I use the Add Server dialog box to specify the alternative location \\LOSANGELES\DC-CONTRACTS. With this replica set of data, users can access the files in the CONTRACTS folder, even if a server goes down. Figure 1 depicts this configuration, showing how two servers can publish shares into a directory within a DFS structure. The dashed line depicts the File Replication Service (FRS), which I will discuss later. The arrows depict the individual servers publishing their shares into the Dfs structure.

Get a Head Start on NT 5.0 with Dfs

Douglass Toombs

(Reprinted from WindowsItPro Magazine)



How Does Dfs Work?

Now that you have a general idea of how to implement Dfs, you can take a look at what happens under the hood. Suppose I want to access the legal document PLEADING.DOC, which resides in the CONTRACTS share on the WASHINGTON server, from an NT 5.0 workstation. From Explorer, I issue the command to connect to \\NETARCHITECT\DFS\.

The NT workstation goes through its standard attempts to locate this resource by trying to attach the various redirectors (such as Server Message Block and NetWare Core Protocol) to the target. However, because a machine called NETARCHITECT doesn't exist in this network, all the redirectors fail. At this point, Dfs, which is part of the Multiple UNC Provider (mup.sys) in NT 5.0, takes over and attempts to resolve the query by checking the Active Directory (AD). The Dfs service on the server responds to the client's request by providing the location of the Dfs root (\\WASHINGTON\DFSROOT) and attaches to that location.

Because Microsoft designed the Dfs client to wait until all primary network redirectors fail before handling a request, you can inadvertently bypass Dfs if you're not careful. If you attach directly to a share with the machine's full UNC path instead of the Dfs path, you will end up in the same location, but without Dfs's load balancing and fault-tolerance benefits.

Get a Head Start on NT 5.0 with Dfs

Douglass Toombs

(Reprinted from WindowsItPro Magazine)

After the Dfs service finds the root, it presents a share with the WASHINGTON directory in it. I click the WASHINGTON directory, and the Dfs resolution process begins again. This time, two valid replicas exist for the target share (\\WASHINGTON\CONTRACTS and \\LOSANGELES\DC-CONTRACTS).

When multiple replicas exist, the workstation intelligently determines which replica is closest, based on site information contained in the AD. Dfs knows which server is closest by checking the site information for the user and for the share the user is trying to reach. If that share isn't available, it will proceed down the list of servers until it eventually finds a responding one. When Dfs finds a responding server, it presents the share to the user. This resolution process is automatic and transparent to the user, which is one reason why I don't have to rush into the office at 3 a.m. if a server crashes.

Once I arrive at the final directory (or share, depending on how you want to look at it), I find the file PLEADING.DOC. Because I'm in Washington, D.C., the intelligent replica selection directs me to the WASHINGTON server for this document. (If I were in Los Angeles, it would direct me to that server.) Although you probably agree that replicas are a good idea, you might be wondering about file consistency issues. Having one logical share pointing to data in two different locations is an invitation to disaster, right? Not exactly.

In NT 4.0, Microsoft recommends making information in Dfs shares read-only. This approach solves the problem but at the cost of making Dfs much less functional. In NT 5.0, Microsoft is resolving the consistency problem with a great new feature called FRS.

What Is FRS?

FRS is a standard service in NT 5.0 that will be a key part of setting up a Dfs tree. FRS will automatically replicate access control lists (ACLs) and file contents in selected directories on NTFS 5.0 volumes. (You must use NTFS 5.0 for FRS to work properly.) FRS will truly make Dfs a strategic component of enterprise networks.

FRS keeps directories in sync with each other. FRS monitors the NTFS journal in an NTFS 5.0 volume, watching for file changes that it needs to replicate throughout the enterprise. When FRS finds modifications to replicate, it generates a package containing the changed files and announces this package to the appropriate systems.

If a target system needs the new information, it accesses the package and applies the new files to the directory. You can control how often FRS conducts this replication process.

Other than setting the replication schedule, FRS synchronizes all the data without human intervention. And therein lies another reason why I don't have to rush into the office at 3 a.m. if a server goes down. When I get into the office the next morning and reboot my server, Dfs will get all the FRS packages from the alternative server and apply them accordingly. My system will be brought up to date, and my local clients will start accessing the primary server again by default.

Although FRS is a key tool in NT 5.0's Dfs, you need to be aware of two quirks. First, FRS monitors changes at the file level. If a user changes a few bytes in a 10MB file, FRS will package and propagate the entire 10MB file. So before you implement FRS, you must carefully plan for the types and amount of data you want to replicate, how often you want to replicate it, and the amount of bandwidth between the sites involved in the replication. Because FRS works silently in the background, you can easily forget about the bandwidth it's using. If you decide to replicate every

Get a Head Start on NT 5.0 with Dfs

Douglass Toombs

(Reprinted from WindowsItPro Magazine)

user's home directory to every other server in your enterprise using a 5-minute replication interval, your network might come to a screeching halt as it tries to keep up with synchronization.

The second quirk is that FRS doesn't synchronize file-sharing locks between systems and instead uses a last-writer-wins methodology. If two users simultaneously access the same file on different physical shares, the situation can get sticky. If each user makes changes and then saves the file on the network, the user who happened to save the document last will have his or her changes distributed throughout the network. The other user's changes will be lost, with no notification to that user.

Possible Uses for Dfs

Because data is the most important resource in a network, having a way to distribute that data and make replica copies available at all times opens up a world of possibilities for using Dfs. For example, Microsoft recommends that you point your Internet Information Server (IIS) server to a Dfs volume for your Internet and intranet sites. This setup can offer several advantages:

If the server hosting your pages goes down, Dfs will simply redirect the requests elsewhere (unless that server is running your IIS process, too—then you need a cluster).

If several different departments in your organization maintain your Web site content, you can distribute the HTML directories across machines for each department, yet have them appear as one combined directory for IIS.

If you decide to move some HTML content from one system to another, you don't have to worry about breaking any links because Dfs can hide the fact that you have moved the data.

With adequate planning, you can use Dfs to avoid system shutdowns for users. For example, suppose your accounting department needs to work over the upcoming weekend, but you were planning to shut down the server containing that department's data. If the accounting data is in a Dfs tree, you can make an alternative share of the data on another server in advance and configure FRS to synchronize the directories. In other words, you can shut down the server you need to work on, and the accounting department can work over the weekend. Everybody wins.

You can also use Dfs tree to maintain hot backups of key data areas, as the Washington, D.C. server crash example illustrates. I recommend making backups of key data areas on different servers, instead of making one large master data area for everyone to work on, because Dfs doesn't synchronize file-sharing locks between replicas.

Get Dfs

Although Dfs will be a standard component of NT 5.0, you can get the Dfs beta for NT 4.0 on Microsoft's Web site (<http://www.microsoft.com>). Although you won't have file synchronization capabilities, Dfs might work just fine for the data in your organization. (Microsoft, for example, uses Dfs to distribute the daily builds of NT 5.0 throughout its enterprise network.) Plus, you'll be one step ahead in the migration to NT 5.0 because Microsoft plans to develop a tool that will let you migrate existing NT 4.0 Dfs components into NT 5.0's Dfs.

So get the Dfs beta for NT 4.0 and install it. Then when your pager goes off at 3 a.m., you'll be able to rest, assured by the knowledge that Dfs is on the job, servicing your critical data.