

Get Going with Dfs

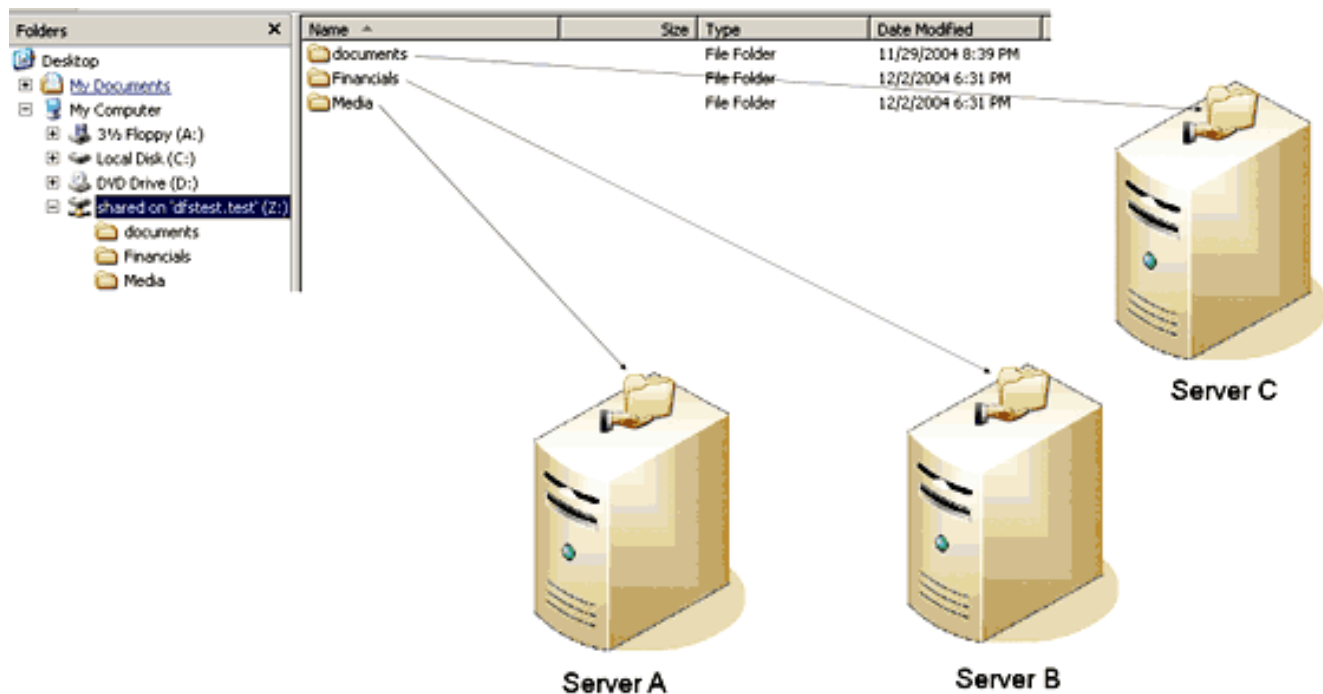
John Savill

(Reprinted from WindowsItPro Magazine)

Microsoft Dfs provides a great way to let users easily access data that's stored on multiple remote computers. Through Dfs, your users can view and access folders as a single set of shares through a familiar, unified folder hierarchy, even when those resources are in different domains or physical sites. If you've resisted using Dfs because you were afraid it might be complex, fear not: Setting up Dfs is a straightforward process, and using it is even easier. I'll help you get started with Dfs by explaining how it works and guiding you through a typical Dfs configuration. After you start using Dfs in your organization, you'll wonder how your users ever functioned without it.

How Dfs Works

The basic element of the Dfs structure is a share that represents the root of the Dfs hierarchy. Through Dfs, these shares form a single, contiguous namespace. Client systems use familiar methods—such as a mapped drive or a Universal Naming Convention (UNC) path—to connect to the Dfs root. After a client connects to the Dfs root, the Dfs structure appears to be a regular share that contains subfolders that users can navigate and search. Each subfolder that's displayed under the Dfs root is actually a link to a share (a link target) anywhere on the network. Dfs automatically redirects the client that's accessing the share to the data's actual location. As Figure 1 shows, the folders that the user sees represent Dfs's redirection of the user to alternate shares on servers A, B, and C. The link target can be any system that uses a network file system that you can access via a UNC path—such as Windows, Novell NetWare, and UNIX or Linux (i.e., NFS) machines.



Dfs provides two types of roots: standalone and Active Directory (AD)–integrated. These types differ in how they store Dfs data. With standalone roots, the Dfs hierarchy, which consists of the various links to network shares, is stored in the Dfs server's local registry. Because of the way the information is stored, it can't be replicated to other Dfs servers, which means that if the sole Dfs server hosting the Dfs root becomes unavailable, the entire Dfs hierarchy is unavailable to all clients on the network. Should the Dfs server become unavailable, clients can still access shares on the servers directly; they just can't use Dfs to access them. You must use standalone Dfs if your environment doesn't have AD or if the Dfs administrators aren't domain administrators and thus can't be granted the authority (i.e.,

Get Going with Dfs

John Savill

(Reprinted from WindowsItPro Magazine)

given access to the DFS-Configuration object in the domain AD partition's System container) to manage Dfs.

Windows 2000 Server and later also support AD-integrated Dfs (aka domain-based or fault-tolerant Dfs). With AD-integrated roots, Dfs information is stored primarily in AD, although the actual Dfs servers still cache copies of the data in memory to minimize Dfs server requests to domain controllers (DCs), thereby reducing Dfs network overhead. You can use an AD-integrated Dfs root only when the Dfs server is a member of a domain; however, the Dfs server doesn't have to be a DC. Essentially, you should use standalone Dfs when you don't have an AD domain, have more than 5000 links, or have legacy clients on your network. For more information about the differences between standalone and AD-integrated Dfs, see the sidebar "Dfs Differences," page 59.

After you've decided which type of Dfs you'll use, you need to set up links and link targets, which contain the data that Dfs will make available to clients. As I mentioned earlier, a link target is the actual resource to which Dfs redirects the client when it accesses a link. A link can have multiple link targets, which helps to provide load balancing and fault tolerance: If a share on one server is unavailable, Dfs redirects the client to another copy of the data. The actual link target that the client uses depends mainly on the client's site. Essentially Dfs is a site-aware service, which means that, by default, when a link target exists in a client's local site, Dfs directs the client to that local target.

Configuring Dfs

Now that you've got a handle on Dfs essentials, you're ready to start setting up Dfs. Your first task is to create a Dfs root. You have two methods for doing this: either by using the Microsoft Management Console (MMC) Distributed File System snap-in or by executing the `dfsutil.exe` command-line utility. Here we'll use the snap-in method, which is a bit easier than the command-line tool when you're just getting started with Dfs. As you become familiar with Dfs, you'll probably want to use `dfsutil.exe`—for example, in a script that populates your Dfs hierarchy with links. Note that on Windows Server 2003, Standard Edition and Win2K Server, a server can host only one Dfs root. Windows Server 2003, Enterprise Edition and Windows Server 2003, Datacenter Edition can host an unlimited number of Dfs roots.

To create a new Dfs root by using the Distributed File System snap-in, perform the following steps:

- Start the Distributed File System snap-in (its shortcut is located in the Administrative Tools Start menu folder).
- Right-click the Distributed File System heading in the snap-in's treeview pane and select New Root (if you're using the Windows 2003 version) or New DFS root (if you're using the Win2K Server version). The remaining steps use the Windows 2003 dialog boxes, although the process is basically the same for Win2K Server.
- At the introduction screen, click Next.
- Select the type of root you want to create (domain or standalone). Click Next.
- If you selected a domain Dfs root, you'll need to enter the name of the domain that will store the Dfs information. If you selected a standalone Dfs root, enter the name of the server that will store the Dfs information. Click Next.

Get Going with Dfs

John Savill

(Reprinted from WindowsItPro Magazine)

- If you selected domain in step 4, you're now asked to select the server that will host the Dfs root. Select a server and click Next.
- Enter the name for the new root and any comments to help identify the root, then click Next. When you enter the root name, you'll also see this name displayed as the share name and a preview of its corresponding UNC path, as Figure 2 shows. For example, for a domain-based Dfs share, the pathname is domain name\share. If the share doesn't currently exist, you're prompted to select a local folder on the machine to use for the share. This share doesn't store any real data; instead, it contains link-type objects that point to where the data physically resides. Select a folder to use for the share and click Next.
- At the confirmation window, click Finish.

New Root Wizard

Root Name
You must provide a unique name for each DFS root.

Type a unique name for this root.

Root name:
shared

Preview of UNC path to the root:
\\dfstest.test\shared

Comments:
Top level shared data DFS root

Share to be used:
shared

< Back Next > Cancel

At this point, clients can connect to the Dfs namespace by using the UNC path \\dfstest.test\shared; they don't need to know anything about which servers are hosting Dfs. Clients running Windows NT 4.0 with Service Pack 6a (SP6a) or later can connect to a domain-based Dfs namespace. However, clients running Windows 98 can access standalone Dfs namespaces but must have the AD client extension installed to access a domain-based namespace. Microsoft Windows Preinstallation Environment (WinPE) instances can access only standalone Dfs namespaces.

Get Going with Dfs

John Savill

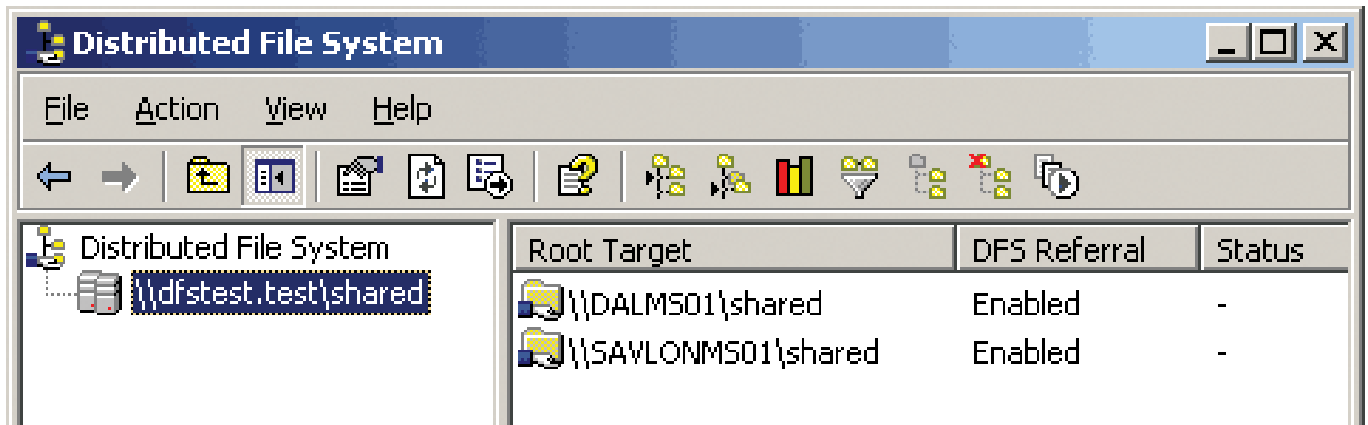
(Reprinted from WindowsItPro Magazine)

To benefit from a domain-based Dfs namespace's fault-tolerance capability, you need at least two Dfs servers to host the Dfs namespace. Follow these steps to configure a second Dfs host server:

- In the Distributed File System snap-in, right-click the root you created and select New Root Target.
- Enter the server name that will be the additional Dfs host for the namespace. Notice that the name of the share (e.g., shared) that Dfs will use to host this copy is set and can't be changed. Click Next.
- If the share name doesn't exist on the target server, you're prompted to select the folder to use, or you can create a new folder on the server, then select it. Make your selection and click Next.
- At the summary dialog box, click Finish.

The Dfs root will now display multiple servers that act as root targets for the namespace, as Figure 3 shows. Clients can now connect and will be directed to one of the Dfs namespace root targets.

However, users who access a root target will see an empty folder because you haven't defined any links yet. The next step, then, is to add some links and link targets that will redirect clients to useful content.



At this stage, to finish your Dfs configuration, you need to create a list of the shares in your organization, determine which data is the same on the different shares, and decide how you want the data to be known to the client (i.e., the folder name and comment). After you've identified this information, you can create the links by performing these steps:

- Right-click the Dfs root and select New Link from the context menu.
- Enter the name of the link (i.e., the folder that clients will see) along with the share to which the link will redirect the client. (You can change this name or add names later.) You can also enter a comment and specify the amount of time that clients cache this redirection information before they query the Dfs server, as Figure 4 shows.
- Click OK.

Get Going with Dfs

John Savill

(Reprinted from WindowsItPro Magazine)

New Link ? X

Enter a name and path for the new link. When users open this link, they are redirected to the shared folder you select as the target.

Link name:
documents

Preview of UNC path to the link:
\\DFSTEST\\shared\\documents

Path to target (shared folder):
\\Dalms01\\documents

Comments:
Main Documentation area

Amount of time clients cache this referral in seconds:
1800

Now when clients access the Dfs namespace, they'll see a folder. When users open the folder, they'll be redirected to a share and will see the content that's stored under the share.

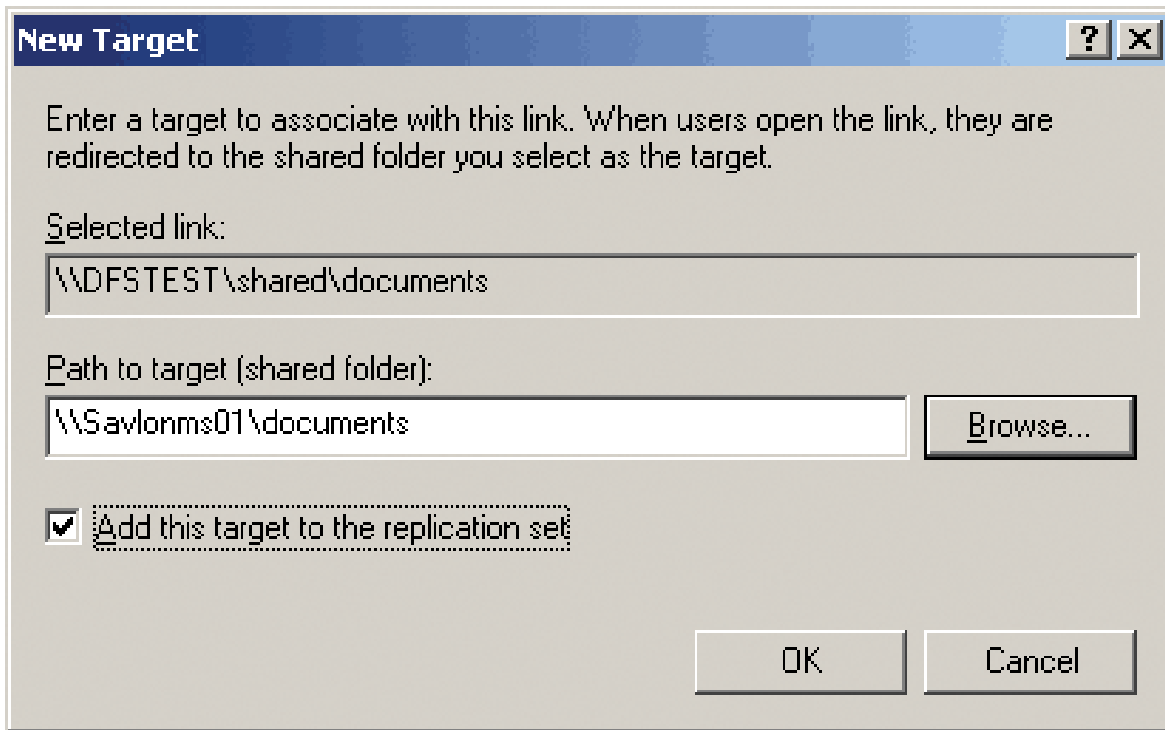
Say you also have a documents folder on a server in a remote office. Instead of creating a separate link to the folder (e.g., LondonDocuments), you can add another link target to the existing link. Setting up multiple link targets is another means for providing fault tolerance. If one link target fails, Dfs can redirect clients to an alternate copy of the data. To add another link target to an existing link, follow these steps:

- Right-click the link and select New Target from the context menu.
- Browse to the new share that will also be a target for the link. You can optionally select the Add this target to the replication set check box, as Figure 5 shows.
- Click OK.

Get Going with Dfs

John Savill

(Reprinted from WindowsItPro Magazine)



When you view your link, you'll notice that two link targets that are enabled. When clients browse to this link, Dfs directs them to one of the targets. You can now repeat the previous steps to configure all the links and targets you require to populate your Dfs structure.

As you've seen, multiple link targets can exist for a link. This capability poses an obvious question: Won't the content be different on the different link targets, meaning that Dfs could randomly redirect clients to different link targets, and thus the clients would see different files? Because multiple targets for a link are effectively separate shares on separate servers, no mechanism exists for keeping the targets' contents synchronized. Therefore, it's entirely possible for the various link targets to have different content, so that a client could browse to a folder, access data, return to the same folder later, but be redirected to an alternate link target and see an entirely different set of data. (However, this scenario is unlikely, as I explain in the Web-exclusive sidebar "Fine-Tuning Dfs Redirection" at <http://www.windowsitpro.com>, InstantDoc ID 45621.) Fortunately, Win2K Server and later Dfs implementations include File Replication Service (FRS), which DCs use to keep their Sysvol shares synchronized. Dfs uses FRS to synchronize the targets of a link that's part of a domain-based namespace. FRS provides various replication options, such as continuous replication, which allows replication of changes in near real time, and replication at certain times of the day. (Windows 2003 R2 will include a brand-new version of FRS just for Dfs.) I provide the steps for configuring Dfs-based file replication in "Setting Up Dfs-Based File Replication." If you have standalone Dfs and require synchronization, you need to use a file-synchronization tool such as the Windows resource kit Robocopy utility to provide that capability.

Dfs Demystified

As you've seen, Dfs greatly simplifies access to network resources for end users and, with AD enabled, provides a measure of fault tolerance. To make sure Dfs works optimally for your organization, you'll need to decide what files need to be replicated and, if necessary, tweak Dfs's redirection. I've covered the essential information you need to get started with Dfs.

Get Going with Dfs

John Savill

(Reprinted from WindowsItPro Magazine)

Setting Up Dfs-Based File Replication

If you have multiple link targets and want to keep files synchronized among the targets, you'll need to configure Dfs-based replication. To set up replication for a link, perform these steps:

- Right-click the link and select Configure Replication.
- At the Configure Replication Wizard's introduction screen, click Next.
- You're prompted to select a target as the initial master of the replication. If you have one share that currently holds the data you want to be replicated to other shares, choose that share as the initial master. Click Next.
- You'll need to select the topology that the replication will use. The default is a ring topology, which is fine for most environments. If your network environment is more advanced, you might want to consider using one of the other topology options, which include hub-and-spoke, full-mesh, and custom. Your topology selection should correspond to your WAN network topology; ideally, your FRS replication should match the network layout. For example, if your network includes a central office and multiple branch offices connected to it, the hub-and-spoke topology would be your best choice. Click Finish.

The link targets will now replicate changes automatically, keeping their content in sync. However, depending on where the servers are geographically located, updates might occur on a delayed schedule. Factors affecting the delay include available bandwidth, the amount of modified data being replicated, the FRS topology you're using, and the replication schedule.

One caveat about Dfs replication: It isn't designed to be used with data that changes frequently in multiple locations or in cases where a file might be updated at the same time on different link targets. FRS doesn't merge file changes together; the most recently saved file is the one that's replicated. Therefore, you should use FRS only to replicate static data, such as templates or company policies or data that will be changed at only one location at a time. FRS's main benefit is that it provides a highly available set of data to protect you in the event of a single-server failure.

Fine-Tuning Dfs Redirection

As I mentioned earlier, Dfs is a site-aware service, which means that when a client accesses a link and multiple link targets are available for that link, Dfs first attempts to redirect the client to a link target in the client's local site. If Dfs can't find an available local target, it redirects the client to another, randomly selected, link target. Dfs's redirection strategy reduces the use of WAN bandwidth when a local link target is available.

Dfs's default behavior of redirecting a client to any alternative link target when it can't find a local link target can be inefficient. For example, if Dfs can't find a local target in Dallas, it might redirect the client to a target in London, although another link exists in New Orleans over a faster connection. However, you can adjust your Dfs configuration so that Dfs redirects clients more efficiently. One option is to configure Dfs so that it redirects clients only to link targets in the users' local sites. You enable this option—called Restricted Same-site Target Selection mode—by executing the Dfsutil command and specifying the /insite parameter, like this:

```
dfsutil /root:\\<server> <dfs root> insite /enable
```

Get Going with Dfs

John Savill

(Reprinted from WindowsItPro Magazine)

The downside of Restricted Same-site Target Selection is that when Dfs can't find a local target, users simply can't access the resource.

Alternatively, if you're running Windows 2003 on your DCs and Dfs servers, you can enable Least-Expensive Target Selection mode. In this mode, if a local-site link target is unavailable, Dfs redirects the client to a link target that's the lowest cost in terms of the amount of bandwidth Dfs uses to access the AD site-link definitions. Least-Expensive Target Selection mode minimizes the usage of slower links and lets clients access shares more quickly. To enable Least-Expensive Target Selection, run the command

```
dfsutil /root:\\<server> <dfs root> sitecosting /enable
```