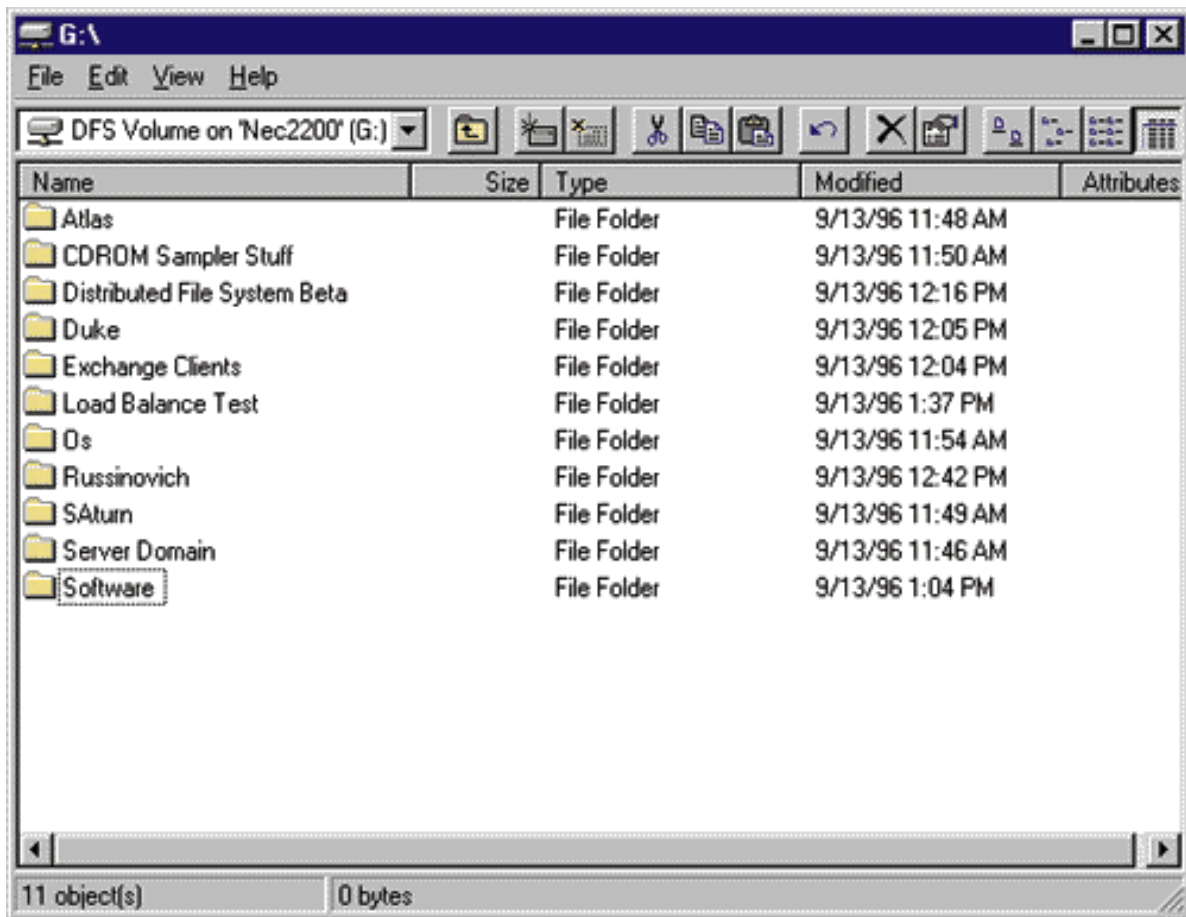


Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)

The Microsoft Distributed File System (Dfs) for Windows NT Server is a utility, currently in public beta, that lets you create a single, hierarchical view of your network's physical resources. Don't confuse Dfs with other distributed file systems used in UNIX environments (for information about the differences see "Dfs vs. DFS,"). With Dfs, you can build a Dfs name space (or directory tree) so users view only one directory that spans all the file servers and server shares in the network, instead of a long list of servers and shares, each with a separate directory. You can position each network resource in the most logical place in your Dfs tree, regardless of where it is actually located in the network. Furthermore, Dfs is only software. No new file systems are created, so no extra security is required beyond native NT security. Screen 1 shows an example Dfs tree created in the Windows NT Magazine Lab.



Dfs adds a layer of abstraction to the physical \\server\share universal naming convention (UNC) name, so you can access network resources in a more intuitive manner. For example, suppose you need to find your human resources department's benefit information. Which search is easier: looking in \corpsrv3\hr\allinfo\pubinfo\1996\benefits (where you first have to hunt around for the server and share names), or looking in \\ntdfs\dfsroot\hr\benefit info (where you can search down the \dfsroot directory tree if necessary)?

Dfs Benefits

Dfs benefits are numerous. System administrators and designers will like how simple it is to consolidate server shares into a single tree that's easy to maintain. You can centrally manage a Dfs tree by controlling rights to the Dfs servers, and transparently add storage where it's needed. Users

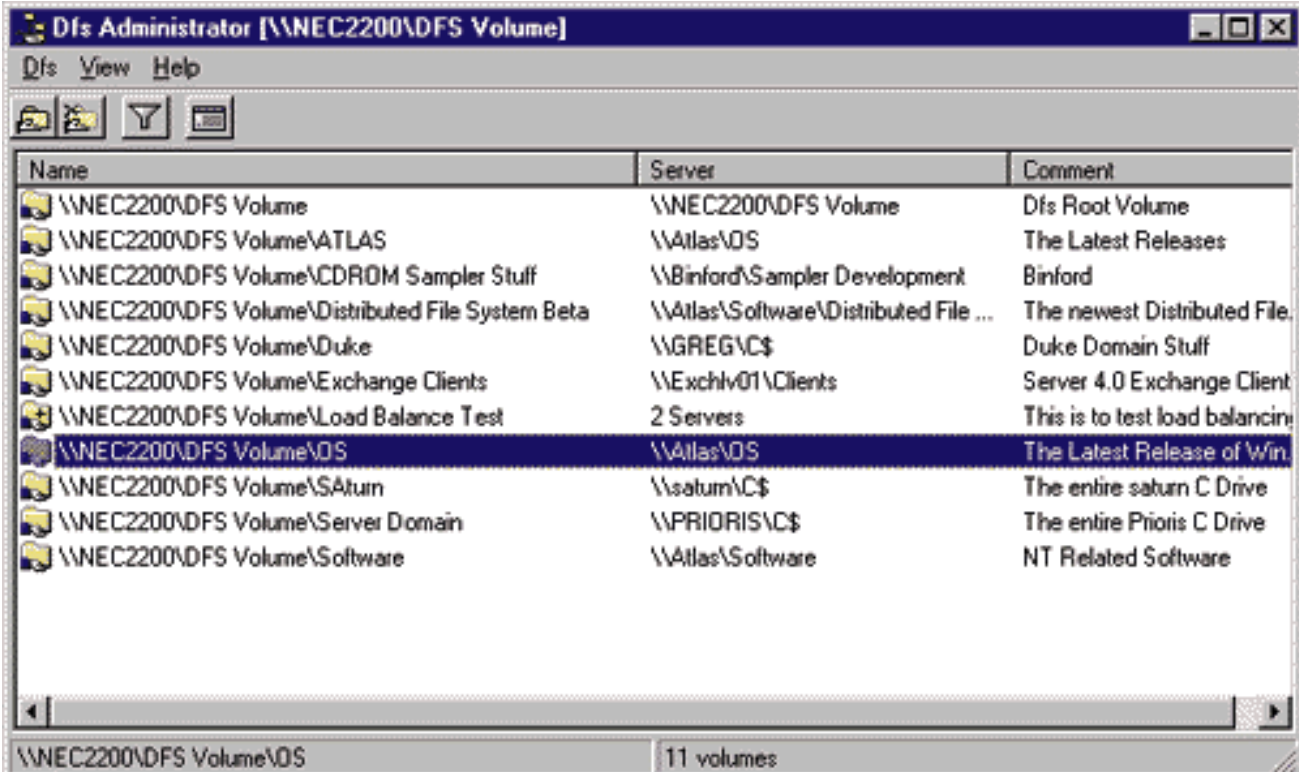
Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)

can connect to one or two Dfs network shares and easily locate data throughout the network without having to remember server or share names--you can assign logical, descriptive names to resources in the Dfs tree regardless of their names in the network. Because a Dfs tree is a virtual representation of physical shares, you can also move the physical location of each share without affecting what the user sees. The Dfs administrator needs to know where the new share physically resides, but your users access the data exactly as before.

Suppose you need to take down a server to upgrade it, but also keep its data available to users. With Dfs you can simply move the data from one server to another, point the Dfs tree at the new physical location (using the Dfs Administrator program shown in Screen 2), and upgrade the original server at your convenience.



The screenshot shows the 'Dfs Administrator' window for '\\NEC2200\DFS Volume'. The window contains a table with three columns: 'Name', 'Server', and 'Comment'. The table lists various Dfs volumes and their corresponding physical server locations. The 'Dfs Volume' \\NEC2200\DFS Volume\OS is currently selected.

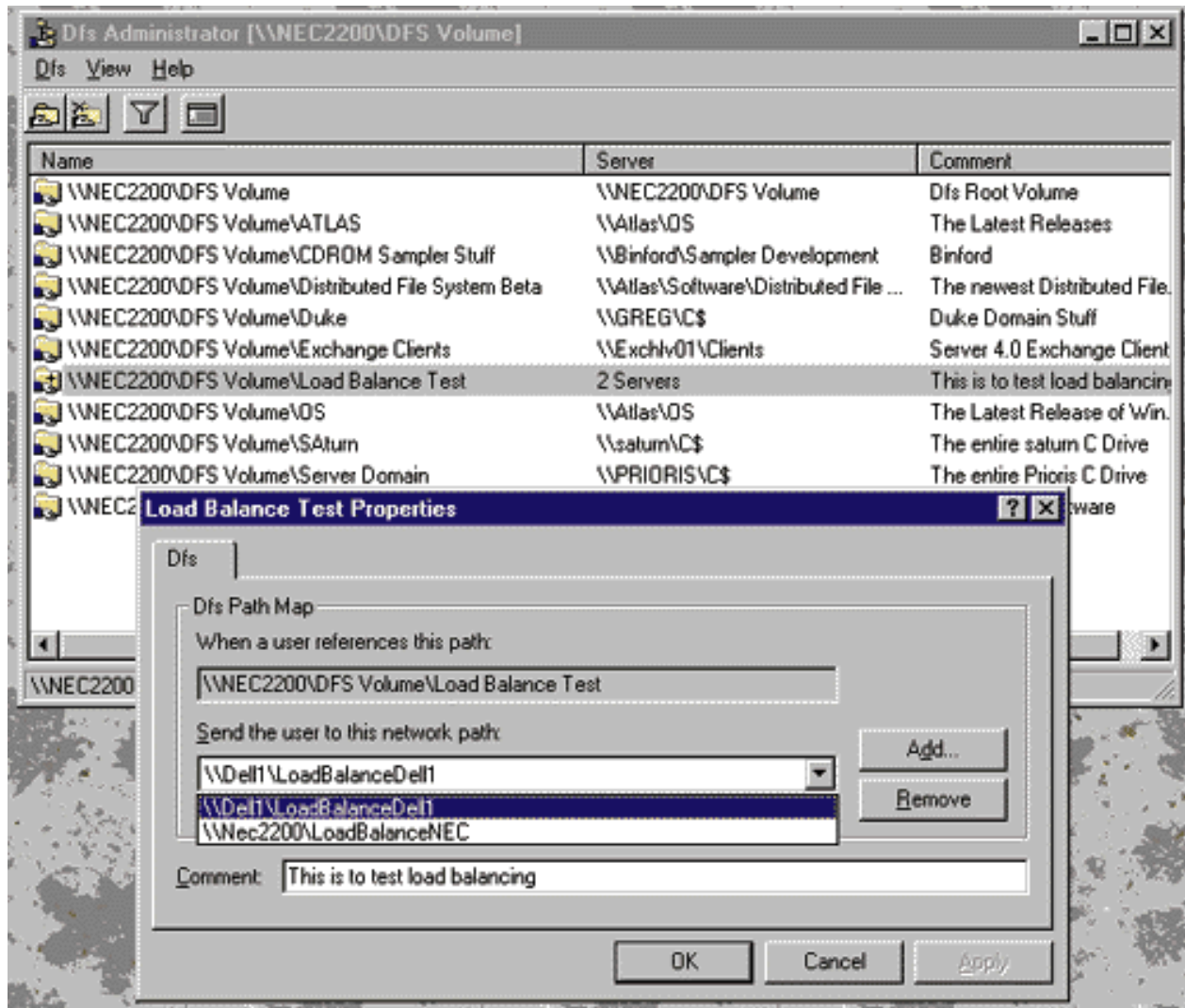
Name	Server	Comment
\\NEC2200\DFS Volume	\\NEC2200\DFS Volume	Dfs Root Volume
\\NEC2200\DFS Volume\ATLAS	\\Atlas\OS	The Latest Releases
\\NEC2200\DFS Volume\CDROM Sampler Stuff	\\Binford\Sampler Development	Binford
\\NEC2200\DFS Volume\Distributed File System Beta	\\Atlas\Software\Distributed File ...	The newest Distributed File...
\\NEC2200\DFS Volume\Duke	\\GREG\C\$	Duke Domain Stuff
\\NEC2200\DFS Volume\Exchange Clients	\\Exchlv01\Clients	Server 4.0 Exchange Client
\\NEC2200\DFS Volume\Load Balance Test	2 Servers	This is to test load balancing
\\NEC2200\DFS Volume\OS	\\Atlas\OS	The Latest Release of Win...
\\NEC2200\DFS Volume\SAturn	\\saturn\C\$	The entire saturn C Drive
\\NEC2200\DFS Volume\Server Domain	\\PRIORIS\C\$	The entire Priors C Drive
\\NEC2200\DFS Volume\Software	\\Atlas\Software	NT Related Software

A Dfs volume can increase data availability because you can use multiple servers (a.k.a. alternate paths) as duplicate storage points for any part of the Dfs tree. Any share in the Dfs tree can point to any number of duplicate data sources. Dfs doesn't replicate the data among the multiple data sources (so you need to use NT's directory replication services or other products to do that), but it does transparently distribute the load (shown in Screen 3). Every user is routed to a different physical resource in a round-robin fashion: The first user goes to duplicate resource A, the next user goes to duplicate resource B, and so on. Each user accesses the same data on physically different servers represented as one virtual directory. (All these maneuvers, of course, are transparent to the user.) With this capability, Dfs provides some measure of fault tolerance: If any of the duplicate resources fail, as long as at least one still functions, users can continue to access the data. Unfortunately, however, Dfs doesn't calculate connection costs; if you have a pair of alternate paths and one is across the country, Dfs will not favor the local path over the long-distance path.

Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)



Can your company benefit from Dfs? Before you spend any time planning what great things you can do with Dfs, you need to see whether your network can use it. Because Dfs uses an updated version of Microsoft's Server Message Block (SMB), you must have the newest network redirectors:

- Your servers must be running NT Server 4.0 to run the Dfs service, but Dfs will access existing network shares, called downlevel volumes, located on any server currently on your network. The servers that contain these ordinary network shares don't have to run NT Server 4.0; they can run NT 3.51, Win95, NetWare, LAN Manager, or even Windows for Workgroups (WFW) 3.11. If the Dfs server can establish a connection to the network share, the Dfs service can use it.
- Your clients must be running either NT Workstation (NTW) 4.0 or Win95 with the updated network redirector (included in the Dfs release).

At this time, Dfs does not fully support NTW 3.51, Windows 3.1, and WFW 3.11 as clients. Amazingly, you cannot view or modify NTFS permissions anywhere along the Dfs tree. You even cannot view permissions for files that reside on a network share you ordinarily could modify if you'd connected using conventional methods. This shortcoming destroys the illusion of a seamless directory tree for

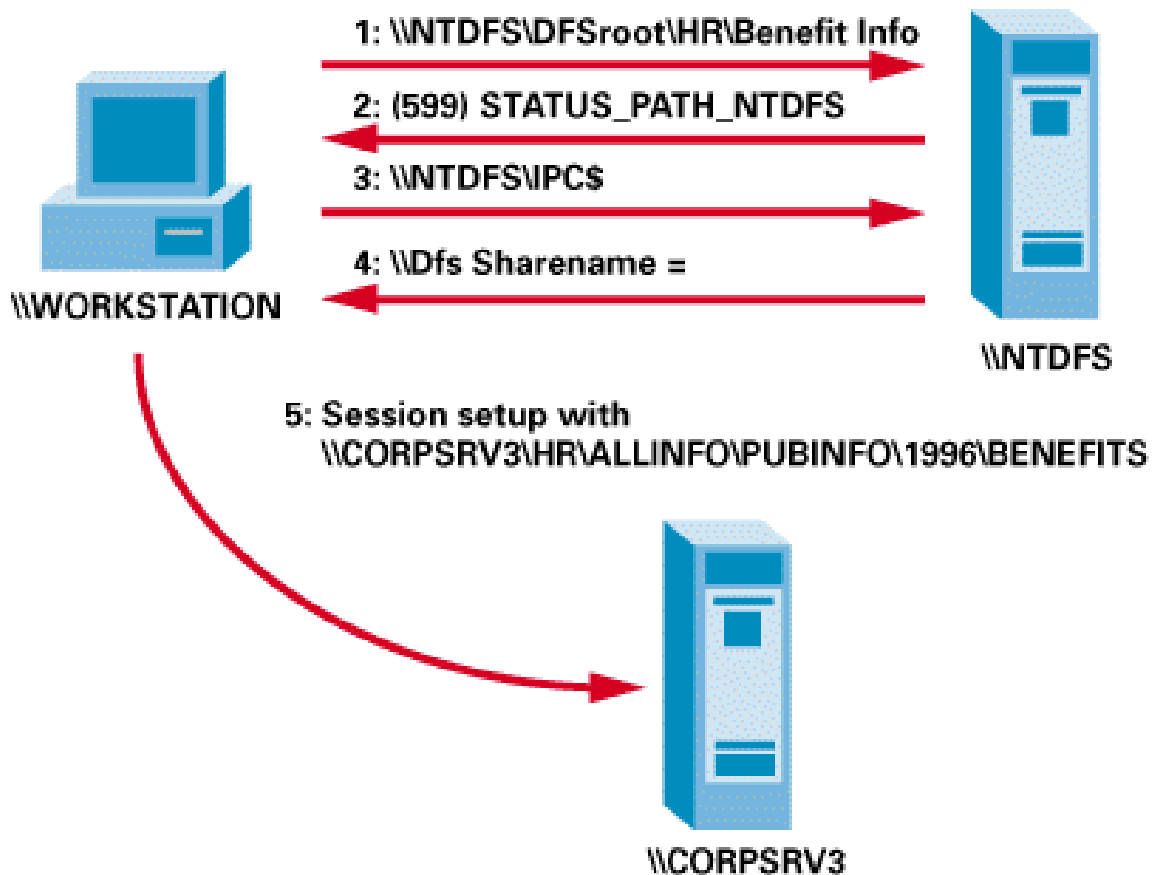
Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)

users who must access the resource outside Dfs to modify its permissions. So Dfs won't be as useful in situations where Dfs users need to modify the data's NTFS permissions. This limitation shouldn't be a problem for resources where data is accessed by Dfs users but controlled by an administrative group that uses conventional network shares.

Thus, how your company benefits from Dfs depends on how many Win95 and NTW 4.0 clients your network includes. If your company plans to upgrade its clients to NTW 4.0 or Win95 in the near future, you can deploy a well-designed Dfs name space now so that it's ready and waiting as users upgrade. As clients upgrade to Win95 and NTW 4.0, they immediately can access the Dfs name space and take advantage of the benefits of Dfs (and entice other users to upgrade).



How Dfs Works

Although you don't need to know how Dfs works to design a Dfs name space, a working knowledge of Dfs is a great help when you troubleshoot your network. Figure 1 shows a typical Dfs session. The following steps provide a behind-the-scenes look at how it works. (You can also view this process with Network Monitor 1.2, which can parse the Transact 2 Data SMB, as shown in Figure 2).

A user with an NTW 4.0 workstation, imaginatively named `\workstation`, attempts to access a Dfs server share named `\ntdfs\dfsroot\hr\benefitinfo`. The `\benefit info` directory actually resides in `\corpsrv3\hr\allinfo\pubinfo\1996\benefits`.

The Dfs server returns a new SMB error code to `\workstation`:

Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby
(Reprinted from WindowsItPro Magazine)

(599)STATUS_PATH_NTDFS

This error code translates to, "Sorry, this is a Dfs path! Get the path information from \\ntdfs\ipc\$ instead."

```
\\workstation obediently connects to \\ntdfs\ipc$.
\\ntdfs sends a Transact 2 Data SMB to \\workstation:
Transact 2 NT Get Dfs Referral
This SMB contains the following data field Dfs Version Referral Information,
which is the actual network share to which \\workstation must connect:
Dfs Sharename = \CORPSRV3\HR\ALLINFO\PUBINFO\1996\BENEFITS
```

This SMB translates to, "Here's the network share you need to connect to."
\\workstation initiates a session directly with \\corpsrv3\hr\allinfo\pubinfo\1996\benefits and begins data transfer. Although our Dfs user is connected to \\corpsrv3, the user thinks the data is coming through the Dfs tree.

FIGURE 2: Parsing the Transact 2 Data SMB

```
Src MAC Dst MAC Protocol Description
WORKSTATION NTDFS SMB C check path, File = \NTDFS\dfsroot\hr\benefit info
NTDFS WORKSTATION SMB R check path - NT error, System, Error, Code = (599)
STATUS_PATH_ NTDFS
WORKSTATION NTDFS SMB C tree connect & X, Share = \\NTDFS\IPC$
NTDFS WORKSTATION SMB R tree connect & X, Type = IPC
WORKSTATION NTDFS SMB C transact2 NT Get DFS Referral
NTDFS WORKSTATION SMB R transact2
SMB: R transact2 NT Get DFS Referral (response to frame 15)
+ SMB: SMB Status = Error Success
+ SMB: Header: PID = 0x2660 TID = 0x0805 MID = 0x0700 UID = 0x0803
SMB: Command = R transact2
SMB: Transaction data
SMB: DFS Path Consumed = 54 (0x36)
SMB: DFS Number of Referrals = 2 (0x2)
+ SMB: DFS Server Function = 6619250 (0x650072)
SMB: DFS Version 2 Referral
SMB: DFS Version Number = 2 (0x2)
SMB: DFS Server Type = Unknown Server Type
SMB: DFS Proximity = 0 (0x0)
SMB: DFS TimeToLive = 300 (0x12C)
SMB: DFS Filename = \NTDFS\dfsroot\hr\benefit info
SMB: DFS 8.3 Filename = \NTDFS\dfsroot\hr\benefi-1
SMB: DFS Sharename = \corpsrv3\hr\allinfo\pubinfo\1996\benefits
SMB: DFS Version 2 Referral
SMB: DFS Version Number = 2 (0x2)
SMB: DFS Server Type = Unknown Server Type
SMB: DFS Proximity = 0 (0x0)
SMB: DFS TimeToLive = 300 (0x12C)
SMB: DFS Filename = \NTDFS\dfsroot\hr\benefit info
SMB: DFS 8.3 Filename = \NTDFS\dfsroot\hr\benefi-1
SMB: DFS Sharename = \corpsrv3\hr\allinfo\pubinfo\1996\benefits
```

Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)

In other words, Dfs tells the client where to go, but it doesn't handle the data transfer. Although all requests to a Dfs share must begin through a Dfs server, the server tells the client which network share to establish a connection with before data transfer begins. This action frees Dfs resources, resulting in a higher overall service capacity for the Dfs server. The user may notice one difference from an ordinary directory tree--the short (you hope) network delay that occurs when the computer connects to \\corpsrv3.

In this example, Dfs could use alternate paths for high-availability load balancing for these heavily accessed shares. In that case, the Benefit Info Dfs share would consist of three or four identical shares on different servers; user requests to access the volume get distributed among the alternate paths. Remember, though, that Dfs 4.0 won't handle synchronization of these volumes, so you have to add a replication step to the process.

Planning a Dfs Name Space

You need to consider several factors when planning a Dfs name space. First, the database that contains the Dfs share information is on only the Dfs server, so you have a single point of failure.

Second, a Dfs server will have high-availability requirements because all Dfs share users must initially go through the Dfs server. Because of its single database and high-availability requirements, a Dfs server is an excellent candidate for a two-node standby cluster. (For more information about clusters, see Mark Smith, "Closing In on Clusters," August 1996). If the Dfs server fails, however, network resources are still accessible by their UNC share names.

Third, because Dfs gives you the opportunity to construct a name space based on logical design rather than physical constraints, you can bypass the inherent limitations that come from trying to organize a company's data based on physical server and share names. You can build a Dfs name space that looks like the organization it serves, and you can easily modify it when reorganizations occur.

Fourth, it's better to build a Dfs server as a domain server rather than a domain controller. Besides removing the load of user authentication, a domain server has local security that can differ from that of the domain in which it resides. With a Dfs domain server, you can tightly control who has rights to create, modify, and delete Dfs directories.

Fifth, the Dfs directory tree above the junction point (i.e., the point where a Dfs directory within the Dfs server points to a real NT share on the network) resides on the Dfs server. If any data is in that tree, it will be on the Dfs server and the server will function like a file server. Clients without the Dfs software (native Win95, WFW, OS/2, NetWare) that access the on-server data won't disconnect from the server like users passing through to get to the junction points. For example, a file in \\ntdfs\dfsroot\hr\benefit info is actually on \\corpsrv3\hr\allinfo\pubinfo\1996\benefits, whereas a file in \\ntdfs\dfsroot\hr resides on the \\ntdfs server.

Last, you can encourage use of Dfs by making new server shares hidden (by placing a dollar sign (\$) in the share name). Once you add the new shares to the Dfs name space, users can access them like any other resource, but they won't be visible for connection outside Dfs.

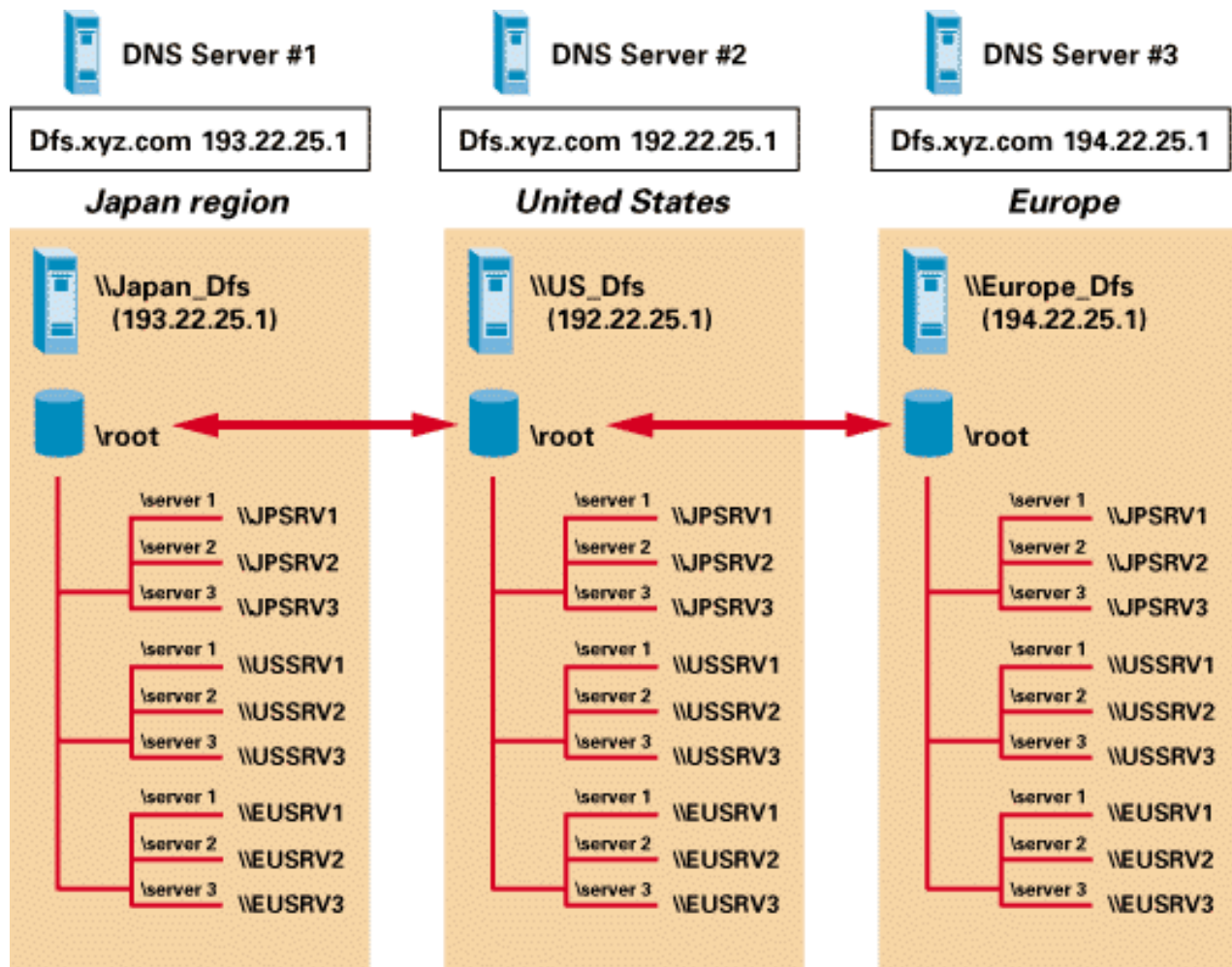
Designing a Corporate Enterprise Dfs Name Space

You can use Dfs to design an enterprise NT name space that avoids some of the limitations of conventional NT domains. Figure 3 shows an example of how to use Dfs to organize the NT shares of a multinational corporation.

Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)



(replicated Dfs structure table)

In this example, the XYZ corporation has three operating regions: Japan, the United States, and Europe. For the sake of simplicity, each region has three servers (xxSRV1, xxSRV2, and xxSRV3), each with one share, that the entire corporation can access. Each region has its own Dfs server, with network addresses 193, 192, and 194, respectively. Each region also has a DNS server that contains the IP address of its region's Dfs server, mapped to a host name dfs.xyz.com. All the clients and servers in each region use only their region's DNS server.

With this design (and the considerations that follow), every user in the XYZ corporation can connect to only one share and access all its servers via an easy-to-understand hierarchy. For example, users in the United States can reach data on \\eusrv2 by drilling down into their corporate-standard Dfs share to \\dfs\root\europa\server2. You can centrally administer the Dfs paths to the resources, while users can still access the shares directly if they want to. Searching for files is easier because massive resources can come under one Dfs root volume (although an indexing server would help avoid networkwide searches). As needed, you can add storage capacity transparently.

To make this design work, the setup must meet several conditions:

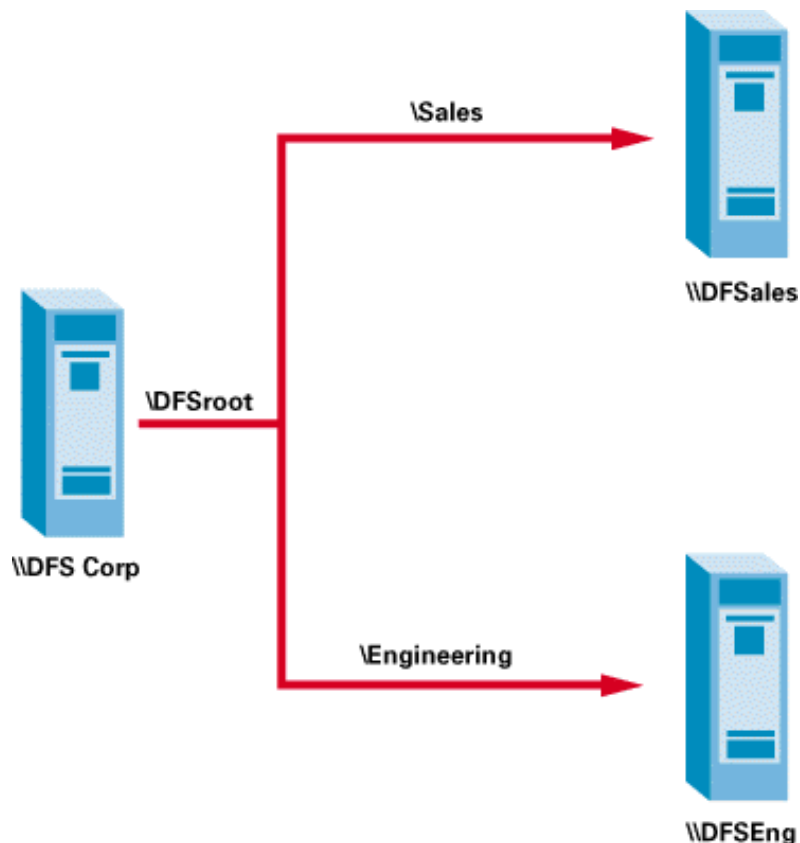
Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)

1. The DNS A record in each server must have the same host name (dfs.xyz.com), but a different IP address--that of the Dfs server running in the local region. This way, everyone in the corporation uses the same name--dfs.xyz.com--to access their local Dfs server. Note that if you set up DNS in this manner, you must disable zone transfers for this record between DNS servers. If you do not, when a zone transfer occurs from DNS #1 (Japan) to DNS #2 (United States), the unique A record for \us Dfs in DNS #2 will be overwritten with the A record for \japan Dfs.
2. You must replicate the Dfs database manually among all the Dfs servers so that they all have identical tree structures. Because dfscmd.exe (the Dfs command-line administration tool) does not currently support an unload or load function, you must create your own. One possible method is a batch file that performs a dfscmd/view\<dfs server>\<dfs root>\ full>dfs.dat to retrieve the Dfs tree data, then parses the tree data and reloads it to another Dfs server via dfscmd/map. Any changes to Dfs tree structures on one server will then be mirrored by Dfs when the database is loaded to the other two servers.
3. Only one junction point is allowed down a directory path in a Dfs server, unless that junction point is directed to another Dfs server. For example, the junction point for the Dfs path \dfs\root\japan\server1 is at \server1, where it redirects to a share on the \jpsrv1 server. If \jpsrv1 is a Dfs server, \server1 can point to another Dfs tree with its own junction points.

Figure 4 shows a design for small businesses. Each department in the company has a Dfs server for its users, and the corporate division has a Dfs server with junction points to the department servers. Corporate employees can view the entire company's server resources through their Dfs server, and sales and engineering department employees usually access only their own resources.



Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)

The Big Picture

Dfs takes the concept of disk file systems--the logical naming and accessing of data on the physical disk--and applies it to networks. Dfs provides a standard naming convention and mapping for collections of servers, shares, and files. Dfs has limitations now, but you can assume Dfs's capabilities will expand to integrate with the Active Directory, the Microsoft Management Console tool, Domain Dfs, and other emerging Cairo technologies. ("Dfs: One Piece of the New Directory Service Solution" shows what part Dfs plays in the Active Directory, NT's new directory service solution.)

Dfs: One Piece of the New Directory Service Solution

Figure A shows the part Dfs plays in NT's new directory service solution, called the Active Directory.

Active Directory: Said to combine the best features of DNS and X.500, the Active Directory will store all accounts, groups, domains, security, and so forth.

Browser (DSWeb): The Directory Service Web (DSWeb) interface will let an administrator manage the Active Directory from a Web browser, locally or remotely. DSWeb will receive information from Internet Information Server (IIS), which will service Active Pages (HTML and ActiveX pages created on the fly). IIS will use the Active Service Protocol (ASP) to communicate with the Active Directory.

Microsoft Management Console: Microsoft Management Console (MMC), previously known as Slate, is a framework for management applications. Microsoft and other vendors will provide snap-ins that let NT administrators manage almost any part of NT from this one interface.

LDAP: Lightweight Directory Access Protocol (LDAP) will let NT's Directory Service (NTDS) talk to any other directory system that supports LDAP through wldap32.dll. This feature also will let the Active Directory manage a non-NT directory service.

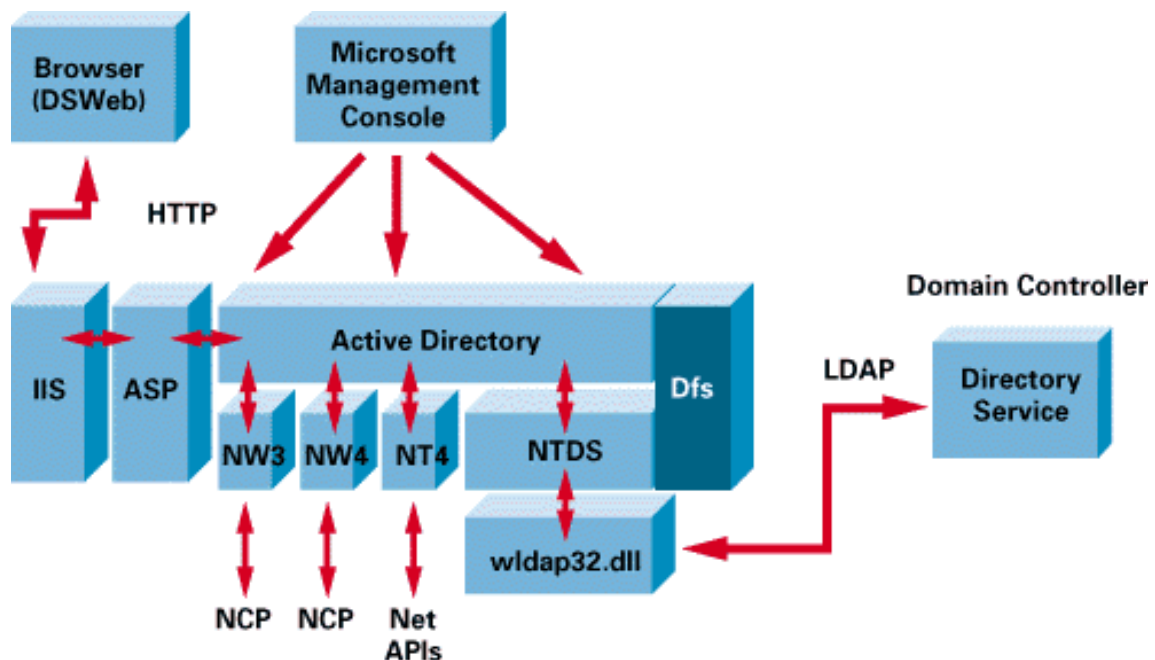


FIGURE A

Dfs: A Logical View of Physical Resources

Tim Daniels and Sean Deuby

(Reprinted from WindowsItPro Magazine)

NT4: The Active Directory will be able to coexist with NT 4's directory service, allowing an easy migration path to the new solution. You will be able to convert your primary domain controller (PDC) to support the new features, but it will appear as a regular PDC to existing backup domain controllers (BDCs) and clients.

NW3 and NW4: The Active Directory will be able to communicate directly with NetWare (NW3, NW4) through NetWare's Core Protocol (NCP).