**1. Describe the idea of the Distribute File System Namespace (DFS-N). Provide details and different scenarious.**

**1.1 Description:**

**Distributed File System** (**DFS**) is a set of client and server services that allow an organization using [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) servers to organize many distributed [SMB](http://en.wikipedia.org/wiki/Server_Message_Block) file shares into a [distributed file system](http://en.wikipedia.org/wiki/Distributed_file_system). DFS provides location transparency and redundancy to improve data availability in the face of failure or heavy load by allowing shares in multiple different locations to be logically grouped under one folder, or DFS root.

Microsoft's DFS is referred to interchangeably as 'DFS' and 'Dfs' by Microsoft and is unrelated to the [DCE Distributed File System](http://en.wikipedia.org/wiki/DCE_Distributed_File_System), which held the 'DFS' trademark[[1]](http://en.wikipedia.org/wiki/Distributed_File_System_(Microsoft)" \l "cite_note-1) but was discontinued in 2005.

It is also called "MS-DFS" or "MSDFS" in some contexts, e.g. in the [Samba](http://en.wikipedia.org/wiki/Samba_(software)) [user space](http://en.wikipedia.org/wiki/User_space) project.

There is no requirement to use the two components of DFS together; it is perfectly possible to use the logical namespace component without using DFS file replication, and it is perfectly possible to use file replication between servers without combining them into one namespace.

A DFS root can only exist on a server version of Windows (from [Windows NT 4.0](http://en.wikipedia.org/wiki/Windows_NT_4.0) and up) and [OpenSolaris](http://en.wikipedia.org/wiki/OpenSolaris)[[2]](http://en.wikipedia.org/wiki/Distributed_File_System_(Microsoft)#cite_note-2) (in [kernel space](http://en.wikipedia.org/wiki/Kernel_space)) or a computer running [Samba](http://en.wikipedia.org/wiki/Samba_(software)) (in [user space](http://en.wikipedia.org/wiki/User_space).) The Enterprise and Data Centre Editions of Windows Server can host multiple DFS roots on the same server. OpenSolaris intends on supporting multiple DFS roots in "a future project based on Active Directory (AD) domain-based DFS namespaces".[[3]](http://en.wikipedia.org/wiki/Distributed_File_System_(Microsoft)#cite_note-3)

There are two ways of implementing DFS on a server:

* *Standalone DFS namespace* allow for a DFS root that exists only on the local computer, and thus does not use [Active Directory](http://en.wikipedia.org/wiki/Active_Directory). A Standalone DFS can only be accessed on the computer on which it is created. It does not offer any fault tolerance and cannot be linked to any other DFS. This is the only option available on Windows NT 4.0 Server systems. Standalone DFS roots are rarely encountered because of their limited utility.
* *Domain-based DFS namespace* stores the DFS configuration within Active Directory, the DFS namespace root is accessible at \\domainname\<dfsroot> or \\fq.domain.name\<dfsroot>. The namespace roots do not have to reside on domain controllers, they can reside on member servers, if domain controllers are not used as the namespace root servers, then multiple member servers should be used to provide full fault tolerance.

**DFS namespaces[[edit](http://en.wikipedia.org/w/index.php?title=Distributed_File_System_(Microsoft)&action=edit&section=1" \o "Edit section: DFS namespaces)]**

Traditional file shares,associated with a single server, have [SMB](http://en.wikipedia.org/wiki/Server_Message_Block) paths of the form \\<SERVER>\<path>\<subpath>. Domain-based DFS file share paths are distinguished by using the domain name in place of the server name, in the form \\<DOMAIN.NAME>\<dfsroot>\<path>. When a user accesses such a share, either directly or by [mapping a drive](http://en.wikipedia.org/wiki/Drive_mapping), their computer will access one of the available servers associated with that share, following rules which can be configured by the network administrator. For example, the default behaviour is that users will access the closest server to them; but this can be overridden to prefer a particular server.

If a server fails, the client can select a different server transparently to the user. One major caveat regarding this flexibility is that currently-open files will potentially become unusable, as open files cannot be [failed-over](http://en.wikipedia.org/wiki/Failover).

**DFS replication[[edit](http://en.wikipedia.org/w/index.php?title=Distributed_File_System_(Microsoft)&action=edit&section=2" \o "Edit section: DFS replication)]**

Early versions of DFS used Microsoft's [File Replication Service](http://en.wikipedia.org/wiki/File_Replication_Service) (FRS) which provides basic file replication capability between servers. FRS identifies changed or new files, and copies the latest version of the entire file to all servers.

[Windows Server 2003 R2](http://en.wikipedia.org/wiki/Windows_Server_2003_R2) introduced "DFS Replication" (DFSR) which improves on FRS by only copying those parts of files which have changed ([remote differential compression](http://en.wikipedia.org/wiki/Remote_differential_compression)), by using data compression to reduce network traffic, and by allowing administrators flexible configuration options for limiting network traffic with a customizable schedule.

**History[[edit](http://en.wikipedia.org/w/index.php?title=Distributed_File_System_(Microsoft)&action=edit&section=3" \o "Edit section: History)]**

The server component of Distributed File System was first introduced as an add-on to [Windows NT 4.0](http://en.wikipedia.org/wiki/Windows_NT_4.0) Server, called "DFS 4.1",[[4]](http://en.wikipedia.org/wiki/Distributed_File_System_(Microsoft)#cite_note-4) and was later included as a standard component of all editions of [Windows 2000](http://en.wikipedia.org/wiki/Windows_2000) Server. Client-side support is included in Windows NT 4.0 and later versions of Windows.

Linux kernels 2.6.14 and later[[5]](http://en.wikipedia.org/wiki/Distributed_File_System_(Microsoft)" \l "cite_note-5) come with an SMB client [VFS](http://en.wikipedia.org/wiki/Virtual_file_system) called "cifs" that supports DFS.

On Mac OSX DFS is supported natively in [Mac OS X 10.7 ("Lion")](http://support.apple.com/kb/ht4794) onward.

**1.2 Scenarios**

**Many File Servers and File Shares**

It’s a common scenario. Over time, a company ends up with a large number of file servers that were created at different times and by different teams. The users are left to figure out where their files are stored, mapping many drives to file shares and learning complex UNC paths like [\\server23\ffiles](file:///\\server23\ffiles) and [\\finance.asia.company.com\spreadsheets](file:///\\finance.asia.company.com\spreadsheets) and even something like [\\10.1.23.45\scratch](file:///\\10.1.23.45\scratch). To complicate things further, when an old file server is replaced, paths are broken and mapped drives stop working. It can be challenging.

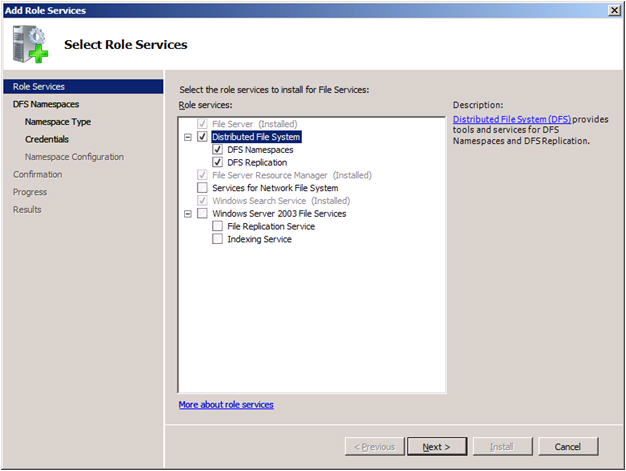
You also probably heard the proposition to invest in a complex project to consolidate everything into a large file server or NAS appliance. While those projects will solve the problem for a while, you will need to spend a lot of resources to move the data around and fix users’ mapped drives and paths. All that to have that new server eventually become too small and need replacing again. Or you will need many of those larger servers, starting the cycle over.

It turns out it doesn’t have to be that way. Windows Server already provides a way to manage a large set of file shares as a consolidated list, under common namespaces. From your users’ perspective, each namespace looks like a single file share with many folders. From an administrators’ perspective, you can easily add more file servers or consolidate existing file servers without complex “forklift” migrations. And those are just a few of the benefits of DFS.

**Adding the DFS Services**

DFS Services are available on all currently supported versions of Windows Server, but there are significant improvements in the Windows Server 2008 editions. The DFS namespace client is available for all currently supported versions of Windows, both client and server. Domain-based DFS namespaces require the use of Active Directory.

To add the DFS Services to Windows Server 2008, you will use the Server Manager tool. First, you need to expand the "Roles" item on the tree to make sure that the File Service role is installed. Then you will right-click on "File Services" and click on option to "Add Role Services". That's where you you will find the option to add DFS:

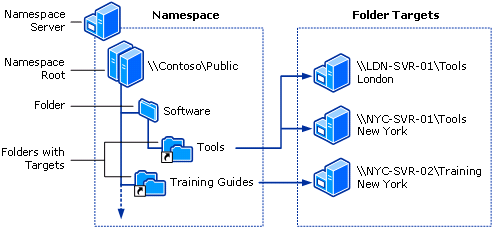


Another way to add the DFS Services is using the ServerManagerCmd.EXE command line tool. You can also use ServerManagerCmd to make sure that the right role and role services are installed. Here's a sample command line:

|  |
| --- |
| C:\>servermanagercmd -query | find "[FS"  [X] File Server [FS-FileServer]  [X] Distributed File System [FS-DFS]  [X] DFS Namespaces [FS-DFS-Namespace]  [X] DFS Replication [FS-DFS-Replication]  [ ] File Server Resource Manager [FS-Resource-Manager]  [X] Services for Network File System [FS-NFS-Services]  [ ] Windows Search Service [FS-Search-Service]  [ ] Windows Server 2003 File Services [FS-Win2003-Services]  [ ] File Replication Service [FS-Replication]  [ ] Indexing Service [FS-Indexing-Service]  C:\> |

**DFS Namespaces**

A DFS namespace is basically a place where you will have links to all your file shares. From an administrator point of view, you should think of it as a folder structure where you keep the list of target file shares. Your users will see it as a single share with many folders and they will have no idea that they are navigating across a set of servers to get to the subfolders and files.



When configuring DFS, you have a choice of using a domain-based or a stand-alone namespace. If you already have Active Directory deployed, you should consider using a domain-based namespace. If you’re not using Active Directory, your only choice is a stand-alone one.

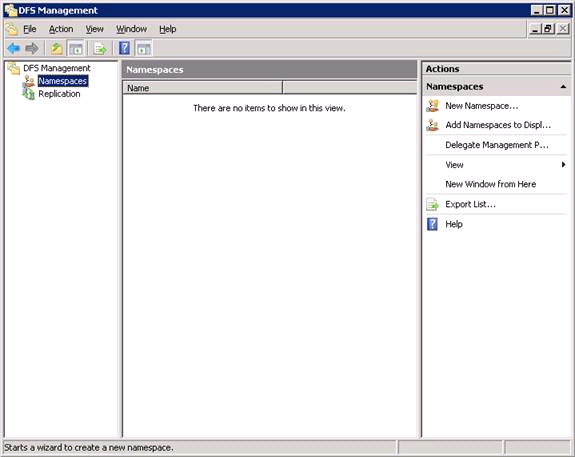
The main advantage of the domain-based namespaces is that your configuration will be stored in Active Directory and you won’t have to rely on a single server to provide the namespace information to your clients. The path users refer to uses the name of the domain and it will not need to change because your namespace server name changed (only if you change your domain name). With a stand-alone DFS, that server name becomes part of the main path to the namespace.

There are also two domain-based DFS modes: Windows Server 2008 mode and Windows Server 2000 mode. Windows Server 2008 mode (which requires Windows Server 2003 forest functional level, Windows Server 2008 domain functional level and Windows Server 2008 running on all namespace servers) includes support for more than 5,000 folders with targets per namespace and access-based enumeration.

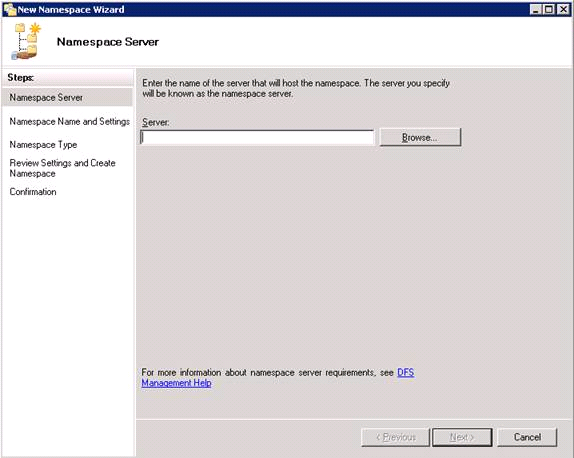
**Creating a Namespace**

After you install the service, your next step is to create the namespace. Here’s are the steps to create a domain-based namespace:

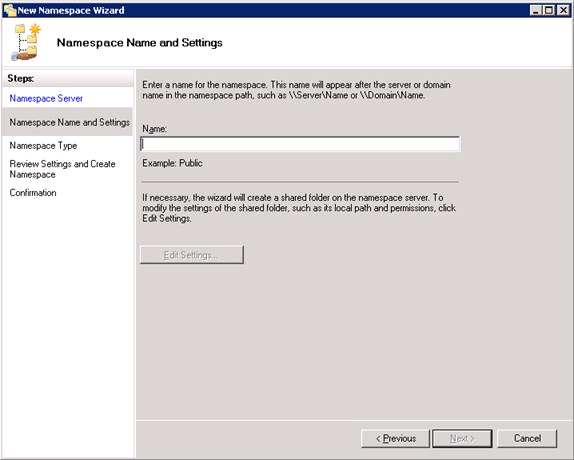
Start “DFS Management” from the “Administrative Tools”. On the tree on the left side, click on “Namespaces”.



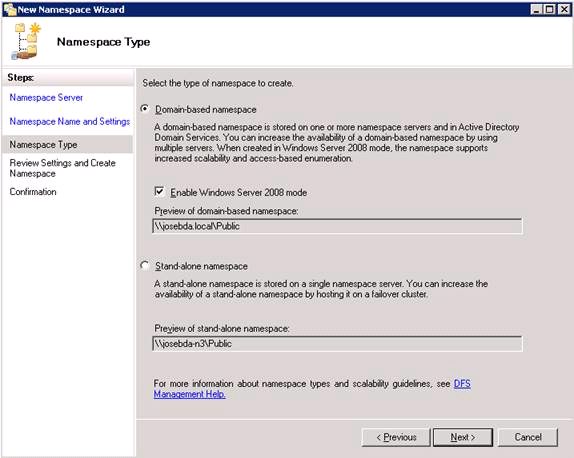
Click on “New Namespace…” action on the panel on the right to start the “New Namespace Wizard”.



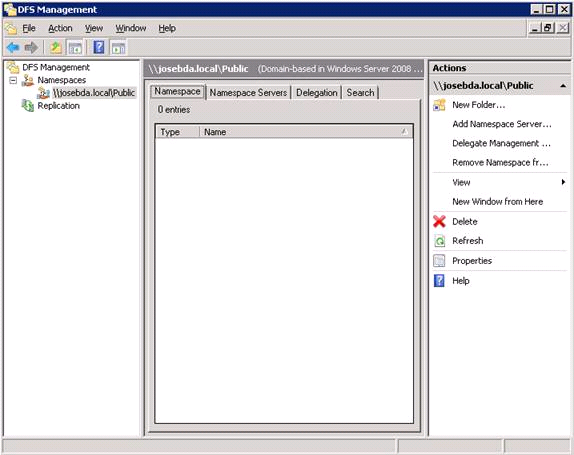
Enter the name of the namespace server and click “Next”.



Enter the name of the namespace and click “Next”.



Select the type of Namespace and click “Next”. In my case, I’m using a domain-based namespace with 2008 mode enabled. On the review screen, click “Create” and then click “Close” when it’s done. You will end up with a brand new namespace with no folders:

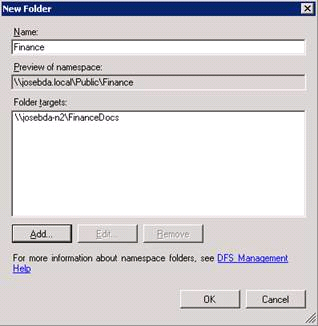


Note: DFS-N “namespaces” are also sometimes called “namespace roots” or simply “roots”. This naming convention was used in older versions of DFS-N and you will also see it in the command-line tools.

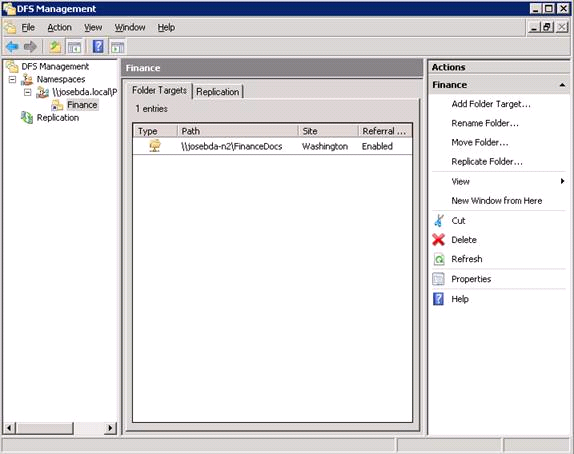
**Adding Folders to the Namespace**

After your create the namespace, you will add folders to it, specifying the associated folder target. This means pointing to the actual file shares, making each one appear to users as a folder under the namespace. Before you do that, you want to think long and hard about the folder structure you’re creating. A basic goal of DFS is to create a stable infrastructure that will not constantly change on your users.

To start, click on the “New Folder…” action on the panel on the right. In the example below, I will enter the name of the folder which as “Finance”. I will also enter one associated folder target, which will be “\\JOSEBDA-N2\FinanceDocs” (this share was configured beforehand). Here are the results right before I click “OK” to create the new folder:



Finally, here’s the end result in the “DFS Management” window:



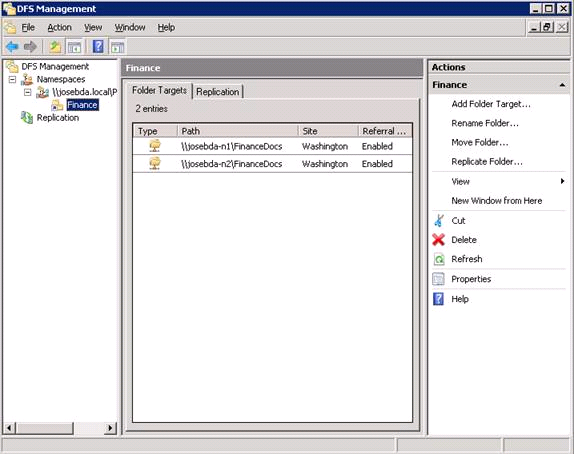
I could also have folders with no targets (just to create a hierarchy in the namespace) or folders with multiple folder targets.

Note: DFS-N “folders” are also sometimes called “links”. DFS-N “folder targets” and also sometimes called “link targets”. These naming conventions were used in older versions of DFS-N and you will also see it in the command-line tools.

**Multiple Targets**

It’s useful to have multiple copies of the same data stored in different file servers. One reason for that is fault-tolerance (if one server is unavailable, you can still access the other one). The other reason is to choose the copy of the data that is closer to you. If you’re in a branch office and you want to access a very large file, you would rather get a copy from a server in that branch.

It’s actually quite simple to add more folder targets to an existing folder or create the folder with multiple targets initially. All you have to do is make sure that you provide the multiple targets for the same folder in that namespace when you configure it.



If you’re using domain-based DFS, the clients will be directed to the target that is closest to them. If there is no target nearby, the clients will be pointed to a remote one. This is similar to what happens when clients are looking for a domain controller. DFS uses the site information in Active Directory to determine which server works best.

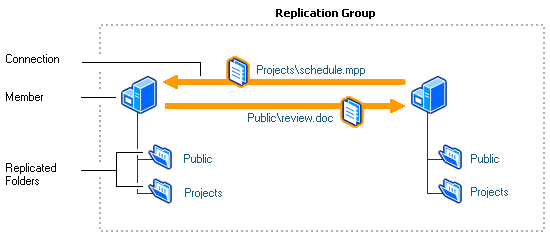
Having multiple file servers serving the same data is also useful for maintenance. If you need to apply a security update or add some memory to the server, you can count on the fact that there is at least another server out there that can service client requests while that specific server is down.

**DFS Replication**

At this point, you’re probably thinking: How does the data get copied across multiple servers?

Well, Windows Server includes a component to replicate data between file servers in case you need that. It’s called DFS-R (Distributed File System Replication). DFS-R was introduced with Windows Server 2003 R2 (replacing the old NTFRS with many advantages). DFS-R can be used for both domain-based and stand-alone DFS.

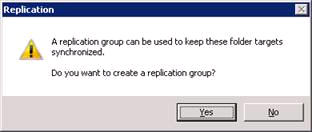
To replicate files between two (or more) shares, you need to create a replication group and specify a few things like which servers will replicate (members) and what they will replicate (replicated folders). DFS-R is aware of site topology. It also has options to control scheduling and the use of bandwidth (throttling). DFS-R uses Remote Differential Compression (RDC), meaning that only changes in the files are sent over the network, not the entire file.



There are specific changes in Windows Server 2008 DFS-R, including improved performance, better network bandwidth utilization, quicker recovery from unexpected shutdowns and support for Read-Only Domain Controllers.

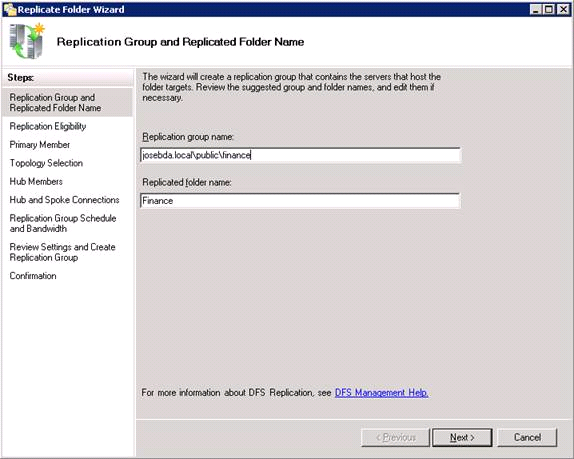
**Configuring Replication**

When you create multiple folder targets for a folder, the DFS Management tool will offer you to automatically setup the replication group:

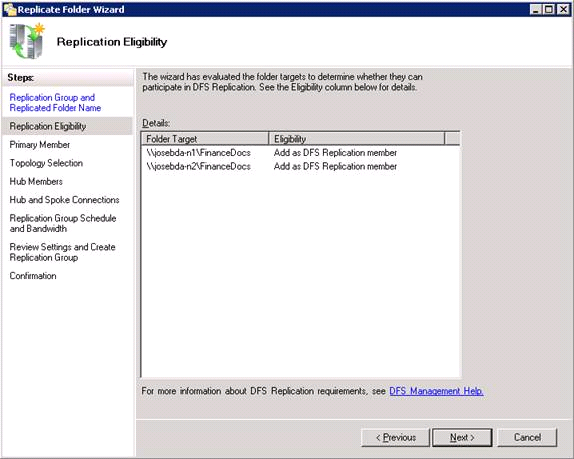


Assuming you have installed the DFS-R role service on both servers, it will go ahead and setup replication for you. It’s done via the “Replicate Folder Wizard”.

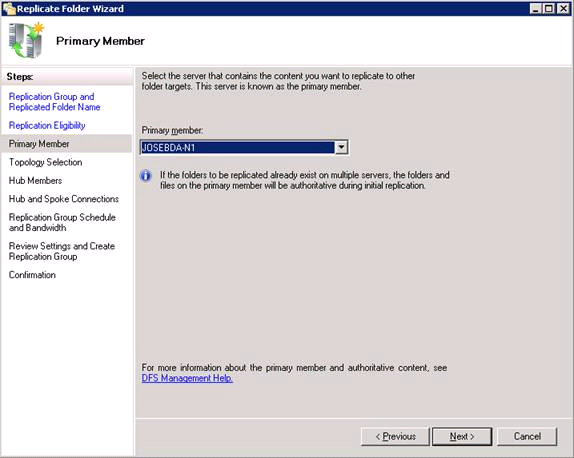
To start, it will ask you the replication group name and the replicated folder name:



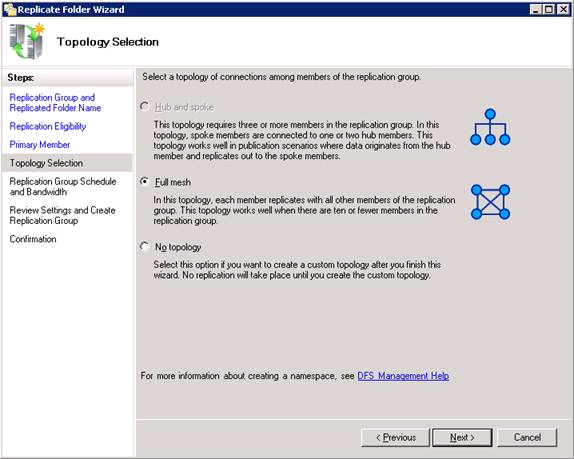
It will then verify if the server working as folder targets for that DFS folder can be a replication member. Our two servers are eligible:



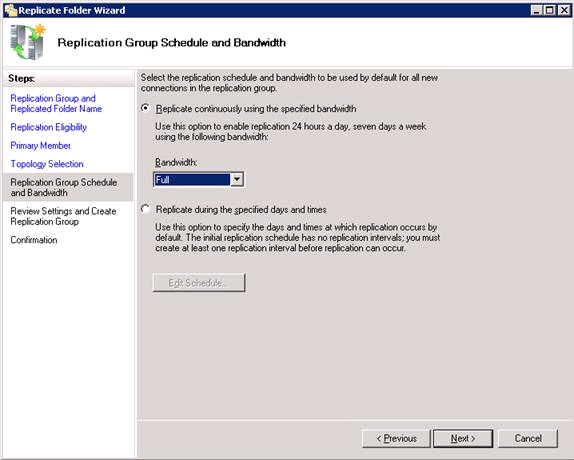
Next, you need to select one of the servers as the primary member. This should be the server that contains the files you want to replicate and will be the authoritative server during the initial replication:



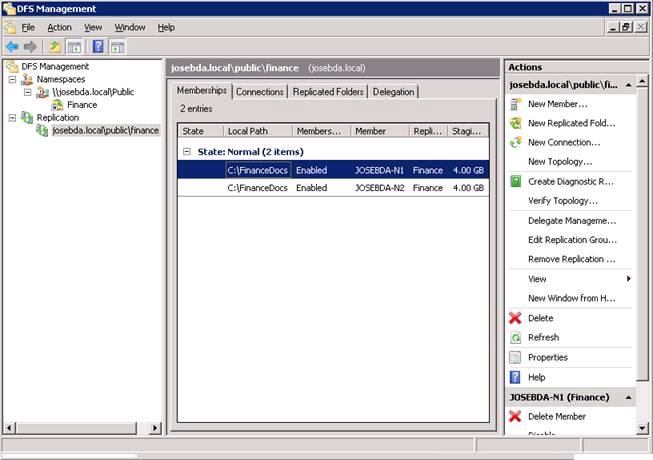
You will also need to select a topology. This is important if you have a larger number of servers and need to control how they replicate. For instance, if you have a central server in HQ and several branch servers, you can make sure the branch servers do not try to replicate with each other directly, but always replicate only with the HQ servers. In our case, with only two servers, "Full Mesh" will do:



Finally, you need to select the bandwidth and schedule. This will allow you do restrict the amount of bandwidth that DFS-R can use and also specify specific times during the day when replication is allowed. In this case, I will keep it simple, replicating at all hours and using the full bandwidth.



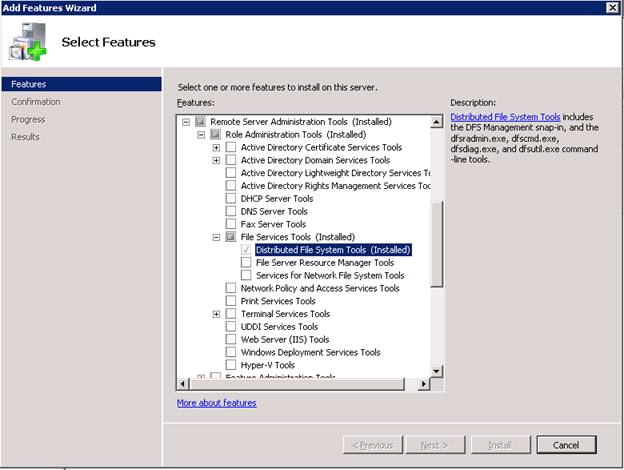
After the review screen, the wizard will set up replication for you. Here’s the end result:



**DFS Tools**

As you saw, the most common administration activities related to DFS-N and DFS-R can be performed using the “DFS Management” MMC. This tool will show up under “Administrative Tools” after you add the DFS role service in Server Manager.

You can also add just the MMC for remote management of a DFS namespace server from a server that does not have the actual DFS Services installed. You will find that option in Server Manager, under Add Feature, Remote Server Administration Tools (RSAT), Role Administration Tools, File Services Tools.



Another way to manage DFS-N is to use DFSUTIL.EXE, which is a command line tool. There are plenty of options and you can automate pretty much any DFS-related activity, from creating a namespace to adding folders to exporting the entire configuration to troubleshooting. DFSUTIL.EXE is an in-box tool in Windows Server 2008. Here’s a sample screen for DFSUTIL:

|  |
| --- |
| C:\>dfsutil  DESCRIPTION: Manages DFS Namespaces, servers and clients.  NOTES: \* Use the Dfsutil /oldcli command to use of the original Dfsutil syntax. \* Use the /? parameter after any command to display help for the command. \* Dfsutil commands use the original Distributed File System terminology,   with updated DFS Namespaces terminology provided as explanation for most   commands. \* If you specify at the end of a command an object (such as a namespace server)   about which you want information, most commands will display the information   without requiring further parameters or commands. For example, when using the   Dfsutil Root command, you can append a namespace root to the command to view   information about the root.  EXAMPLES: Dfsutil Root Dfsutil /oldcli Dfsutil /?  ------ Commands Supported ------ Root                  Displays,creates,removes,imports,exports namespace roots. Link                  Displays, creates, removes, or moves folders (links). Target                Displays,creates,modifies folder targets (link targets). Property              Displays or modifies a folder target or namespace server. Client                Displays or modifies client information or registry keys. Server                Displays or modifies namespace configuration on the server. Diag                  Perform diagnostics or view dfsdirs/dfspath. Domain                Displays all domain-based namespaces in a domain. Cache                 Displays or flushes the client cache.  C:\> |

Note: DFSCMD is an older version of the DFS-N command-line tool. You should use DFSUTIL instead.

For DFS-R, the command line tool is DFSRADMIN.EXE. Here’s a sample screen:

|  |
| --- |
| C:\>dfsradmin  === Supported Objects === You can perform actions on the following objects: Bulk          Perform several actions by using a single input file Conn          Perform actions on connections between members of a replication group Health        Generate a health report for one or more members of a               replication group Mem           Perform actions on a member of a replication group Membership    Perform actions related to a member's participation in a               replicated folder PropRep       Generate propagation report for one or more propagation test files PropTest      Generate and drop a propagation test file on a membership,               used for testing replication RF            Perform actions on a folder that is replicated between members               of a replication group RG            Perform actions on a group of computers that participates in               replication Sub           Perform actions related to a member's subscription in               replication groups  Type "DfsrAdmin <Object> /?" for detailed help. Usage: DfsrAdmin <Object> <Action> [<SubObject>] [<SubAction>]        </Parameter:Value> ... [</Optional-Parameter:Value> ...]        [/Domain:<value>] [/DC:<value>] [/CSV] [/Force]  === Optional Parameters Supported For All Commands === /Domain: Specify the domain for the replication group /DC:     Specify the domain controller to connect to in the replication          group's domain /Force:  Specify that a failed operation is skipped when an action causes a          series of operations to be performed  === Optional Parameter Supported For All List Commands === /CSV:    Dumps the list output in CSV format  C:\> |

Sources:

http://en.wikipedia.org/wiki/Distributed\_File\_System\_(Microsoft)

http://blogs.technet.com/b/josebda/archive/2009/03/10/the-basics-of-the-windows-server-2008-distributed-file-system-dfs.aspx