

What Is DFS (Distributed File System)?



One of the goals of most information technology (IT) groups is to manage file server resources efficiently while keeping them available and secure for users. As networks expand to include more users and servers—whether they are located in one site or in geographically distributed sites—administrators find it increasingly difficult to keep users connected to the files they need. On one hand, distributing resources across a network makes them more available to more people and promotes cross-organizational efforts. On the other hand, storing files on different file servers located throughout an organization makes it difficult for users to know where to look for information. Administrators also find it difficult to keep track of all the servers and all of the people who use those servers. The task of swapping out an old server becomes a major communication chore when users across an organization must be notified to update links and file paths. To help administrators address these problems, Windows Server 2003 includes Distributed File System (DFS). DFS allows administrators to group shared folders located on different servers by transparently connecting them to one or more DFS namespaces. A DFS namespace is a virtual view of shared folders in an organization. Using the DFS tools, an administrator selects which shared folders to present in the namespace, designs the hierarchy in which those folders appear, and determines the names that the shared folders show in the namespace. When a user views the namespace, the folders appear to reside on a single, high-capacity hard disk. Users can navigate the namespace without needing to know the server names or shared folders hosting the data. DFS also provides other benefits, including the following:

Simplified data migration

DFS simplifies the process of moving data from one file server to another. Because users do not need to know the name of each physical server or shared folder that contains the data, administrators can physically move data to another server without needing to reconfigure applications and shortcuts and without needing to reeducate users about where they can find their data. This minimizes the impact of server consolidation on users. It also allows administrators to deploy additional file servers and present the folders on those new servers as new folders within an existing namespace.

Increased availability of file server data

In the event of a server failure, DFS refers client computers to the next available server, so users can always access shared folders without interruption.

Load sharing

DFS provides a degree of load sharing by mapping a given logical name to shared folders on multiple file servers. For example, suppose that \\Company\StockInfo is a heavily used shared folder. Administrators can use DFS to associate this location with multiple shared folders on different servers, even if the servers are located in different sites.

Security integration

Administrators do not need to configure additional security for DFS namespaces because file and folder security is enforced by existing the NTFS file system and shared folder permissions on each target. For example, a user navigating a DFS namespace is permitted to access only the files or folders for which he or she has appropriate NTFS or shared folder permissions.

Common DFS Scenarios

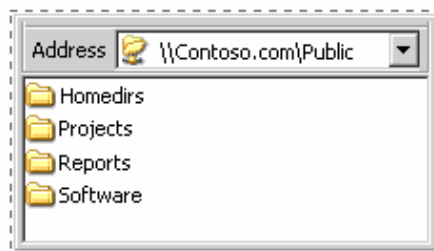
DFS is commonly used in the following scenarios:

Server Consolidation

Many organizations today are consolidating older file servers throughout the organization into fewer, larger, more powerful file servers. Consolidation reduces the cost of managing multiple file servers and increases the efficiency of storage allocation and backup tasks. Organizations that have implemented DFS can

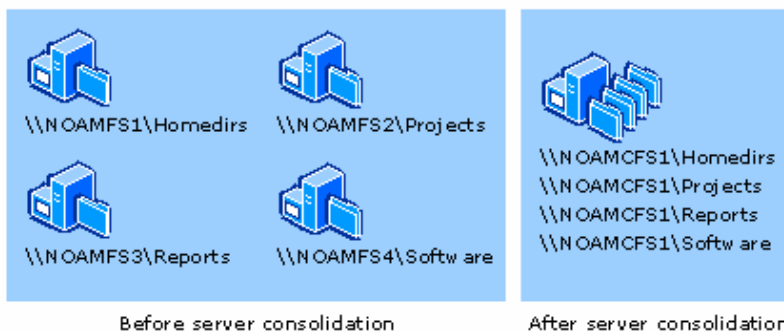
perform server consolidations without impacting the way users' access data. The following figure illustrates this benefit.

How DFS Eliminates the Impact of Server Consolidation on Users



What the User Sees

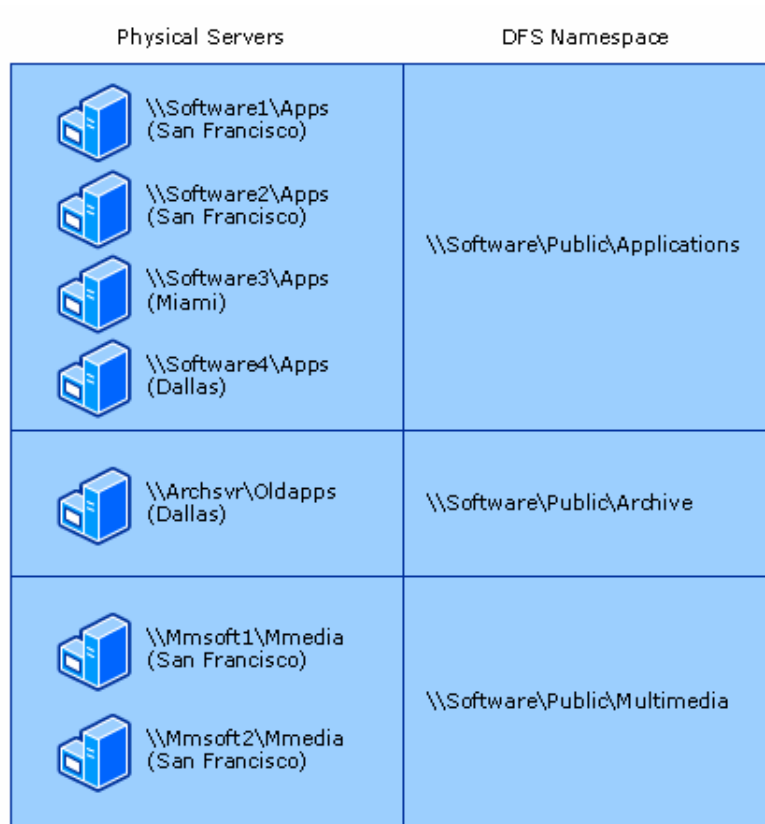
Physical Location of Data



Publishing Applications

DFS is commonly used to publish applications to users throughout the organization. Using DFS in this scenario provides a number of benefits, such as the ability to use multiple servers to host application data and distribute the load across servers. A feature in DFS known as “least expensive target selection” ensures that users are connected to the closest server. The following figure illustrates a DFS namespace used to publish applications in an organization based in San Francisco with offices in Miami and Dallas.

Using DFS to Publish Applications



This organization has three types of software:

- Business-critical software and operating systems that must be available at all times.
- Previous versions of software that are still in use in the Dallas branch office.
- Multimedia software used primarily in San Francisco.

The organization uses four servers to host the business-critical software and operating systems, including two servers in the San Francisco site. Using two servers to host the applications ensures that a failure on one server does not

cause the data to become unavailable. All users can access this software at \\Software\Public\Applications, and users are automatically directed to the server in their site (San Francisco, Dallas, or Miami).

Because the archived software is used only in the Dallas office and the data is not business-critical, only a single server hosts that data. The multimedia software is not business-critical, but the organization uses two servers for this software to improve server response times because the client portion of the multimedia software accesses files from the server.

Increasing Data Availability

As described in the scenario for publishing applications, administrators can use DFS to increase the availability of data by storing the data on multiple servers. DFS makes this process transparent by presenting to users what appears to be a single folder in the namespace. Administrators can use File Replication service (FRS) or some other replication method, such as the Windows Resource Kit Tool Robocopy, to keep the data synchronized on the servers. If one of the servers hosting data is unavailable, clients are referred to another server that hosts the data.

Technologies Related to DFS

File Replication service (FRS) can be used to keep data in DFS shared folders synchronized among servers. However, DFS and FRS are two separate technologies, and DFS does not require FRS. You can use other replication methods, such as manual copying, the Resource Kit Tool Robocopy, or other replication tools to keep DFS shared folders synchronized. Conversely, if you want to use FRS to keep data in shared folders synchronized, you must use DFS.

DFS Dependencies

DFS has the following dependencies:

- Active Directory replication. Domain-based DFS requires that Active Directory replication is working properly so that the DFS object resides on all domain controllers in the domain.

- Server Message Block (SMB). Clients must access DFS root servers by using the SMB protocol.
- Remote Procedure Call (RPC) service and Remote Procedure Call Locator service. The DFS tools use RPC to communicate with the DFS service running on DFS root servers.
- Distributed File System service dependencies. The Distributed File System service must be running on all DFS root servers and domain controllers so that DFS can work properly. This service depends on the following services:
 - The Server service, Workstation service, and Security Accounts Manager (SAM) service on DFS root servers. The Distributed File System service also requires an NTFS volume to store the physical components of DFS on root servers.
 - The Server service and Workstation service on domain controllers

Designing Distributed File Systems

Welcome to the design guide for the Distributed File System solution in the Microsoft® Windows Server™ 2003 R2 operating system. This preliminary guide contains design recommendations for two scenarios, data publication and data collection, in which DFS Namespaces and DFS Replication are commonly used. (Additional recommendations and scenarios will be available as they are developed.) This guide is intended for IT planners and architects who are evaluating these technologies or creating a distributed file system design for their organizations.

If you are not familiar with DFS Namespaces and DFS Replication in Windows Server 2003 R2, we recommend that you read the document titled "Overview of the Distributed File System Solution in Windows Server 2003 R2" available on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55315>). The overview document describes the benefits of DFS Replication and the improvements it offers over File Replication service (FRS). It also describes the DFS Namespaces enhancements exposed in Windows Server 2003 R2. These enhancements, introduced as updated application programming interfaces (APIs) in Windows Server 2003 with Service Pack 1 (SP1), provide easier management and more flexibility for namespaces used in branch offices.

Distributed File System Scenarios and Features

Before you begin your design, it is helpful to understand the scenarios for which these technologies were designed and the basic features that can be configured for DFS Namespaces and DFS Replication in Windows Server 2003 R2. As you review the scenarios, keep in mind that there are no dependencies between DFS Namespaces and DFS Replication. Both services can be used independently of one another, but when used together they can help you achieve more powerful end-to-end scenarios of high availability and WAN load balancing.

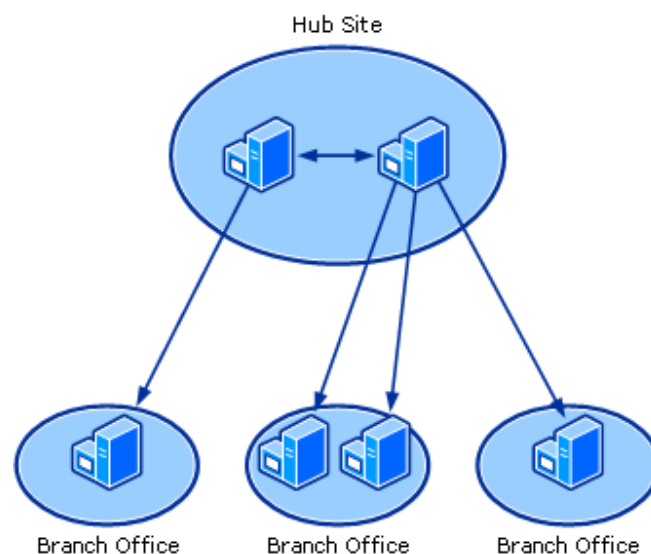
Recommended Scenarios for Using Distributed File System

DFS Namespaces and DFS Replication can be used together to implement the following scenarios.

Data Publication

DFS Namespaces and DFS Replication in Windows Server 2003 R2 can be used to publish documents, software, and line-of-business data to users throughout an organization. In this scenario, data is distributed to multiple servers using DFS Replication, and user access to the data is simplified and made highly available using DFS Namespaces.

The following figure illustrates how DFS Replication can be used to replicate data in a branch office environment where data originates on one or more hub servers in a hub site or data center and replicates to servers in branch offices.



Because DFS Replication is designed for slow WAN links, DFS Replication is ideal for distributing files to branch offices in remote locations. Remote differential compression (RDC) helps reduce the amount of network bandwidth used for replication, and DFS Replication can resume replicating any partially replicated files that result from WAN interruptions. Some additional benefits of using DFS Replication for data distribution are as follows:

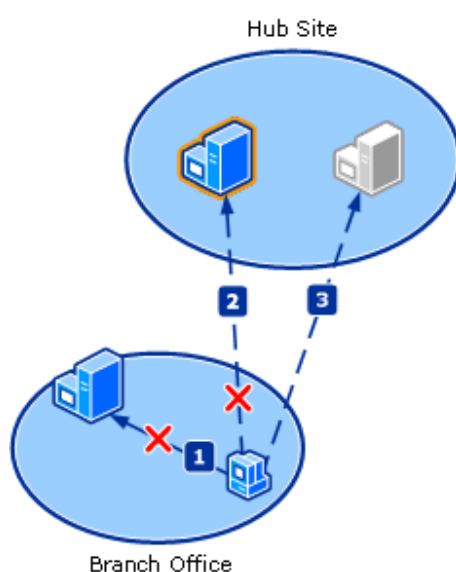
- To reduce the WAN traffic necessary to replicate new files that originate on the hub server, cross-file RDC can be used to identify files that are similar to the file that needs to be replicated. This is useful when a file exists on the hub server but not the branch server. Instead of replicating the entire file, DFS Replication can use portions of files that are similar to the replicating file to minimize amount of data transferred over the WAN. (Requirements for cross-file RDC are described in "DFS Replication Features" later in this guide.)

- To further reduce replication traffic needed for initial replication of new data, you can prestage the branch servers by using a restored backup. DFS Replication can use RDC and cross-file RDC to reduce the bandwidth required to replicate any new files or portions of changed files.
- You can schedule replication to occur during off-hours, and you can set bandwidth throttling to regulate how much bandwidth is used during replication.
- DFS Replication is very efficient in terms of load on the hub server when replicating to a large number of branch servers, because DFS Replication stages the file to be replicated (that is, prepares the file for replication with RDC hashes) once and re-uses the staged file to replicate to all partners. DFS Replication provides numerous additional benefits not described here. For a complete list of these benefits, see the document titled "Overview of the Distributed File System Solution in Windows Server 2003 R2" available on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55315>). Although DFS Replication alone is sufficient to distribute data, using DFS Namespaces allows you to simplify how users access the distributed data and increases the availability of the data. When you create a namespace, you group shared folders located on different servers and present them to users as a virtual tree of folders. Any folder in the namespace can be hosted by multiple servers, each of which holds a replica (kept in sync by DFS Replication) of the published data in that folder. When browsing the namespace, users see a single folder and are not aware that the folder is hosted by multiple servers. The underlying servers are completely transparent to the users, who access the data using a UNC path, such as \\Contoso.com\Software\Products\Microsoft\Office. There are several aspects of DFS Namespaces that work well in branch office environments. For example:
 - Client computers access servers in their own Active Directory site first, if a server is present in the site. You can optionally restrict client computers so that they access only servers in their own site.
 - If a server fails, clients can fail over to another server in the same site (if one exists). If no same-site servers exist, clients can fail over to a server that has the

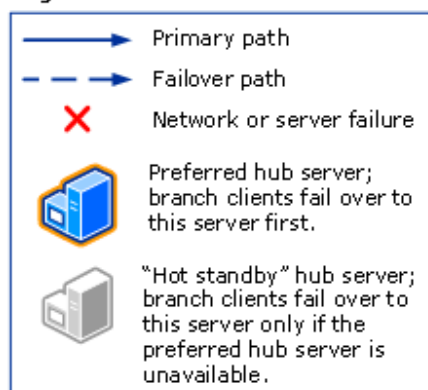
lowest connection cost (as defined in the Active Directory Sites and Services snap-in).

- After the local server is restored, you can configure the namespace so that clients fail back to the local server. (Note that this feature requires that you install a hotfix on the clients.)

The following figure illustrates two features of DFS Namespaces, client failover and target priority, as they might be used in an environment that has a hub site and two branch offices. Assume that during normal operations, the client at the branch office accesses the local server using a referral from the namespace server. If the local branch server fails, and target priority is configured, failover would occur as follows:



Legend

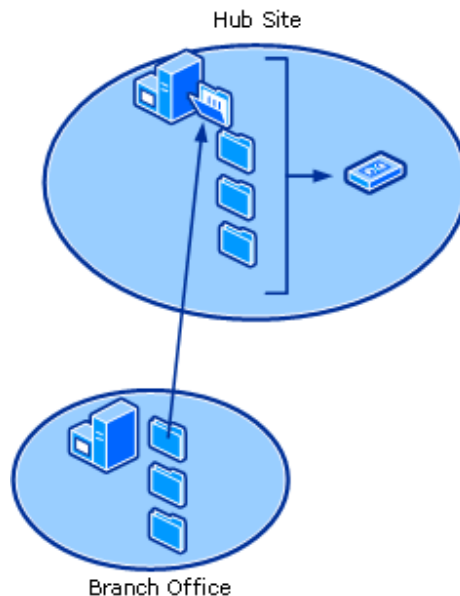


1. The client in the branch site attempts to access a folder target on the local server, but the server is unavailable.

2. The client attempts to fail over to one of the two servers in the hub site. If you want clients to always fail over to a particular server in the hub site, you can configure that hub server's target priority as first among targets of equal cost. The client attempts to fail over to this preferred hub server first.
3. If the preferred hub server is unavailable, the client attempts to access the other hub server, which might be a "hot standby" server for the hub site. You can configure this hub server's priority as last among all targets of equal cost.

Data Collection

The data collection scenario helps address the need to eliminate the use of tape backup in branch offices. To accomplish this, DFS Replication is used to replicate data from a server in a branch office to a server in a hub site or data center. Administrators at the hub site can use backup software to back up the branch server's data from a hub server, eliminating the often error-prone process of having end users performing the backups at branch offices that are not staffed by trained IT personnel. Centralizing backups also helps reduce hardware and operational costs. The following figure illustrates this scenario.

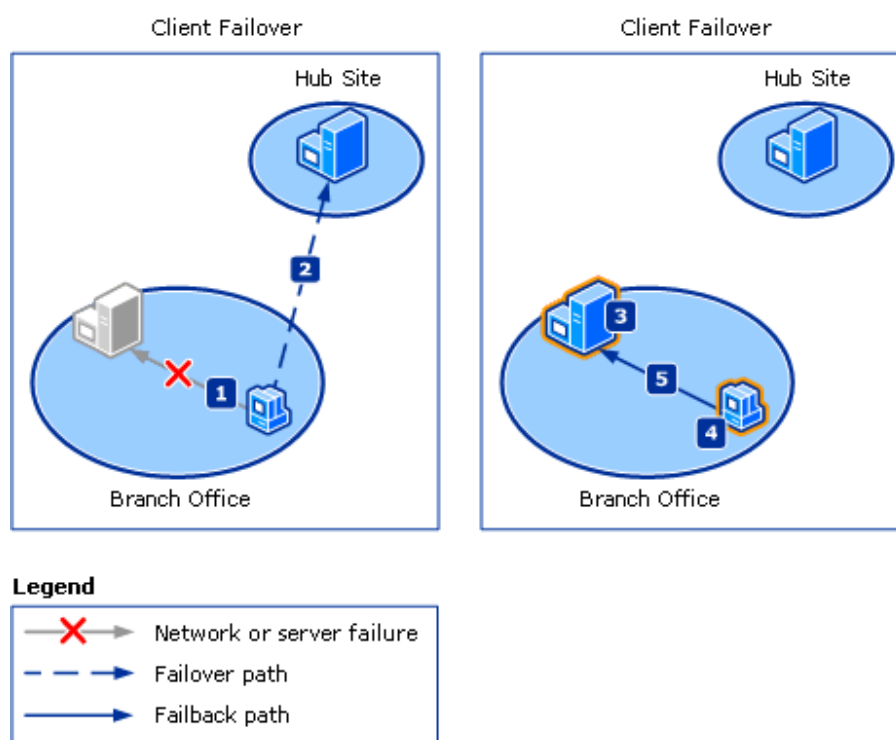


Thanks to RDC, DFS Replication replicates only the differences (or changes) between the two servers; as a result, bandwidth use during replication is minimized, an important consideration for branch offices that use low-bandwidth WAN connections to the hub office. In addition, bandwidth throttling can be used

to regulate the amount of bandwidth used during replication, providing you with more control over WAN traffic.

When DFS Replication is used in conjunction with DFS Namespaces, you can configure a namespace so that branch clients always connect to the branch server. If the branch server becomes unavailable, branch clients fail over to the hub server. And, if client failback is configured, branch clients will fail back to the branch server after it is restored. (Client failback requires a hotfix to be installed on clients. For more information, see "Client compatibility for DFS Namespaces" later in this guide.)

The following figure illustrates the client failover process, which is described in more detail after the figure.



In the previous figure, client failover and failback works as follows:

1. The local branch server or fails or is otherwise unavailable due to network problems.
2. The client fails over to the server in the hub site. The client will access this server until the client's referral expires, the client computer is rebooted, or the client's referral cache is cleared. (A referral is an ordered list of servers that host the shared folders associated with a folder in the namespace.)

3. The branch server is restored.
4. The client requests a new referral after the referral expires, the client is rebooted, or the client's referral cache is cleared. (This step is not related to the branch server's restoration.)
5. After receiving a new referral, the client fails back to the restored branch server.

Depending on your environment, you might consider using Data Protection Manager instead of DFS Replication, especially if you have no other need for replication. We recommend using Data Protection Manager if your connections are greater than 512 kilobits per second (Kbps), all servers are in a single domain, and your organization cannot afford additional storage in the data center as needed by DFS Replication to collect the data and then protect it by Data Protection Manager. When you use DFS Replication to collect data on the hub server and Data Protection Manager to protect the data at the data center, the data in effect is duplicated on the hub server and the Data Protection Manager server.

Features in Distributed File System

To help you design and effectively use a distributed file system in your organization, Distributed File System provides a number of configurable settings and features in Windows Server 2003 R2. These settings and features are briefly described in the following sections.

DFS Replication Settings and Features

The following settings and features in DFS Replication can be customized or enabled as necessary to design a DFS Replication solution for your organization. For a complete list of DFS Replication features and benefits, see "Overview of the Distributed File System Solution in Windows Server 2003 R2" on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55315>).

RDC

Remote differential compression (RDC) is a protocol that can be used to efficiently update files 64 kilobytes (KB) or larger over a limited-bandwidth network. RDC detects insertions, removals, re-arrangements of data in files regardless of file type, enabling DFS Replication to replicate only the changes

when files are updated. To compute the deltas to replicate, RDC typically works on an older version of the file with the same name that exists at the appropriate location in the replicated folder tree on the receiving member. RDC can also use a file with the same name in the Conflict and Deleted folder.

RDC is not used on files smaller than 64 KB; in this case, the file is compressed before it is replicated. You can also disable RDC on connections that are in a LAN where network bandwidth is not contended.

Cross-file RDC

On servers running Windows Server 2003 R2, Enterprise Edition or Datacenter Edition, an additional function of RDC, known as cross-file RDC, can be used to further reduce bandwidth usage. Cross-file RDC uses a heuristic to identify files that are similar to the file that needs to be replicated and uses the similar files as candidates for RDC. Cross-file RDC is useful when a file exists on the sending member and not the receiving member, but similar files exist on the receiving member. Instead of replicating the entire file, DFS Replication can use portions of files that are similar to the replicating file to minimize amount of data transferred over the WAN. Cross-file RDC can use multiple files as candidate files for RDC seed data.

Replication schedule and bandwidth throttling

DFS Replication supports replication scheduling and bandwidth throttling in 15-minute increments during a 7-day period. When specifying a replication window, you choose the replication start and stop times as well as the bandwidth to use during that window. The settings for bandwidth usage range from 16 kilobits per second (Kbps) to 256 megabits per second (Mbps) as well as full (unlimited) bandwidth. You can configure a default schedule and bandwidth that applies to all connections between members and optionally create a custom schedule and bandwidth for individual connections.

Because members of a replication group are often located in different time zones, it is important to consider the time zones of the sending and receiving members when you set the schedule. The receiving member initiates replication by interpreting the schedule either in Coordinated Universal Time (UTC) or in the receiving member's local time, depending on which setting you choose. You can

choose this setting for the replication group schedule and for custom schedules on individual connections.

Replication filters

You can configure file and subfolder filters to prevent files and subfolders from replicating. Both types of filters are set on a per-replicated folder basis. For more information about the types of files that are filtered by default, see "Review DFS Replication Requirements" later in this guide.

Staging folder

DFS Replication uses staging folders to act as caches for new and changed files to be replicated from sending members to receiving members. Each replicated folder uses its own staging folder, and each staging folder has a configurable quota. The quota, which governs when files are purged based on high and low watermarks, must be carefully set based on each replicated folder's replication activity and the disk space available of the server. Guidelines for sizing the staging folder are described in each scenario; general guidelines are also described in "Review additional guidelines and considerations for DFS Replication" later in this guide.

Conflict and Deleted folder

DFS Replication uses a last writer wins method for determining which version of a file to keep when a file is modified on two or more members and each member has not seen the other's version. The losing file is stored in the Conflict and Deleted folder on the member that resolves the conflict. The Conflict and Deleted folder can also be used to store files that are deleted from replicated folders. Each Conflict and Deleted folder has a quota that governs when files are purged for cleanup purposes.

Disabled memberships

A membership defines the relationship between each replicated folder/member pair. Each membership has a status, either enabled or disabled. If you do not want a replicated folder to be replicated to certain members, you can disable the memberships for those members. Doing so allows you to replicate folders to only a subset of replication group members.

DFS Namespaces Settings and Features

The following settings and features in DFS Namespaces can be customized or enabled as necessary to design a DFS Namespaces solution for your organization. For a complete list of DFS Namespaces features and benefits, see "Overview of the Distributed File System Solution in Windows Server 2003 R2" on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55315>).

Referral ordering

A referral is an ordered list of targets, transparent to the user, that a client receives from a domain controller or namespace server when the user accesses the namespace root or a folder with targets in the namespace. The client caches the referral for a configurable period of time.

Targets in the client's Active Directory site are listed first in a referral. (Targets given the target priority "first among all targets" will be listed before targets in the client's site.) The order in which targets outside of the client's site appear in a referral is determined by one of the following referral ordering methods:

- Lowest cost
- Random order
- Exclude targets outside of the client's site

You can set referral ordering on the namespace root, and the ordering method applies to all folders with targets in the namespace. You can also override the namespace root's ordering method for individual folders with targets.

Failover and failback

Client failover in DFS Namespaces is the process in which clients attempt to access another target server in a referral after one of the servers fails or is removed from the namespace. Client failback is an optional feature that enables a client to fail back to a preferred, local server after it is restored.

Failback only occurs when a client has failed over to a more expensive server (in terms of site link cost) than the server that is restored. If the restored server has the same cost as the server that the client is currently connected to, failback does not occur to the restored server. For example, if there are two servers

(Server 1 and Server 2) in the client's site, and Server 1 fails while the client is connected to it, the client will fail over to Server 2. However, the client will not fail back to Server 1 when it is restored, because Server 1 has the same site link cost as Server 2.

For details about the hotfixes and operating systems that the clients and namespace servers must run for failback to work successfully, see "Client compatibility for DFS Namespaces" later in this guide.

Target priority

You can assign a priority to individual targets for a given namespace root or folder. This priority determines how the target is ordered in a referral. The options are:

- First among all targets
- Last among all targets
- First among targets of equal cost
- Last among targets of equal cost

It is important to note that setting target priority on a target will result in that target always being present in a referral, even in cases where you set the **Exclude targets outside of the client's site** option on the folder associated with the target.

Redundant domain-based namespace servers

A domain-based namespace can be hosted by multiple namespace servers to increase the availability of the namespace. Putting a namespace server in remote or branch offices also allows clients to contact a namespace server and receive referrals without having to cross expensive WAN connections.

Root scalability mode

To maintain a consistent domain-based namespace across namespace servers, it is necessary for namespace servers to periodically poll Active Directory to obtain the most current namespace data. If your organization will use more than 16 namespace servers to host a single namespace, we recommend that you enable root scalability mode. When this mode is enabled, namespace servers running Windows Server 2003 do not send change notification messages to

other namespaces servers when the namespace changes, nor do they poll the PDC emulator every hour. Instead, they poll their closest domain controller every hour to discover updates to the namespace. (Regardless of whether root scalability mode is enabled, changes to the namespace are always made on the PDC emulator.)

Although traffic to the PDC emulator is reduced in root scalability mode, the trade-off is that namespace servers running Windows Server 2003 must wait for the hourly polling before they discover changes to the namespace. Therefore, if the namespace changes frequently, the namespace servers might provide out-of-date referrals until they poll for changes.

Feature Requirements for Distributed File System

DFS Namespaces and DFS Replication are two separate technologies and therefore have separate requirements. These requirements are described in the following sections.

DFS Replication Requirements

The basic requirements for DFS Replication are:

- You must first update the Active Directory® schema to install the Active Directory classes and attributes used by DFS Replication. These schema changes are provided on disc 2 of the Windows Server 2003 R2 operating system installation discs. This schema can be applied to domain controllers running Windows Server 2003 R2, Windows Server 2003, and Windows® 2000 Server.
- All servers that participate in replication must run Windows Server 2003 R2.
- All members of a replication group must be in the same forest. You cannot enable replication across servers in different forests.
- Cross-file RDC is available only when the sending or receiving replication partner (in a pair of replicating servers) is running Windows Server 2003 R2, Enterprise Edition, or Windows Server 2003 R2, Datacenter Edition. If a pair of replicating servers are both running Windows Server 2003 R2, Standard Edition, or

Windows Server 2003 R2, Web Edition, cross-file RDC is not used for replication between those members.

The full list of DFS Replication requirements is described in "DFS Replication Requirements" later in this guide. For more information about extending the schema, see the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55329>).

DFS Namespaces Requirements

The requirements for DFS Namespaces vary for each setting or feature. The following table provides a high-level overview of the DFS Namespaces requirements for each configurable DFS Namespaces setting or feature. More in-depth requirements are described in "DFS Namespaces Requirements" later in this guide.

Setting or Feature	Active Directory Requirements	Namespace Server Requirements	Client Requirements
Lowest cost referral ordering	<ul style="list-style-type: none"> Sites must be set up in Active Directory. (Otherwise, all targets have the same cost.) Domain controllers must run Windows Server 2003 to provide domain-based root referrals based on cost. The domain controller acting as the Intersite Topology Generator (ISTG) in each site must run Windows Server 2003 to calculate site costs. 	Namespace servers must run Windows Server 2003 or Windows Server 2003 R2 to provide referrals based on cost.	N/A
Client failback	N/A	Namespace servers must run Windows Server 2003 with SP1 or Windows	Clients must run either: <ul style="list-style-type: none"> Windows® XP with Service Pack 2 and the Windows XP Client

Setting or Feature	Active Directory Requirements	Namespace Server Requirements	Client Requirements
		Server 2003 R2.	Failback hotfix. <ul style="list-style-type: none"> Windows Server 2003 SP1 and the Windows Server 2003 Client Failback hotfix.
Target priority	Domain controllers must run Windows Server 2003 with SP1 or Windows Server 2003 R2 to provide domain-based root referrals based on target priority.	Namespace servers must run Windows Server 2003 with SP1 or Windows Server 2003 R2 to provide referrals based on target priority.	N/A
Redundant domain-based namespace servers ¹	<ul style="list-style-type: none"> Active Directory is required for domain-based namespaces. 	Namespace servers must run Windows 2000 Server or Windows Server 2003. Namespace servers can be member servers or domain controllers.	N/A
Root scalability mode (domain-based namespaces only)	N/A	Namespace servers must run Windows Server 2003.	N/A

¹By default, the Standard Edition of Windows Server 2003 R2 and Windows Server 2003 supports one namespace per server. However, if you install the hotfix described in article 903651 in the Microsoft Knowledge Base, you can create multiple domain-based namespaces on a server running Standard Edition.

Review Settings and Features Used in Each Distributed File System Scenario

The following table compares the recommendations for the configurable settings and features for DFS Namespaces and DFS Replication, based on each scenario covered in this guide. More detailed information about these settings and features is available in the respective scenarios described later in this guide.

Setting or Feature	Data Distribution	Data Collection
DFS Replication	Use to replicate data that originates on the hub server to the branch servers.	Use to replicate data that originates on the branch servers to the hub servers.
Primary member	Choose the hub server as the primary member.	Choose the branch server as the primary member.
RDC	As new files are published, cross-file RDC can be used to replicate new files, based on their similarity to existing files. Cross-file RDC requires the hub or branch server to run Windows Server 2003 R2, Enterprise Edition or Datacenter Edition.	RDC is recommended so that only changes (deltas) are replicated from the branch servers to the hub server. (RDC is enabled by default.)
Replication schedule and bandwidth throttling	Use a staggered schedule to provide better scale-out on the hub server.	Varies; use a nightly replication schedule if you need to conserve bandwidth during the day. If you are using a namespace for failover purposes, consider whether to use a 24x7 schedule so that the hub server will be as consistent as possible with the branch server, if a branch client fails over to a hub server.
Replication filters	Use the default filters.	Add filters to exclude files you do not want to replicate or back up, such as .pst files, .mp3 files, and so forth.
Staging folders	Size the staging folder quota by evaluating the length of the replication window and the amount	To prevent frequent staging folder cleanups while replicating, size the staging folder quota so that it can

Setting or Feature	Data Distribution	Data Collection
	of data to replicate. The goal is to avoid having to restage files for subsequent replication windows.	accommodate staging files for all changes that occur during the day in order.
Conflict and Deleted folders	File conflicts are unlikely because data originates at the hub and is read-only for users. Therefore, file conflicts do not need to be considered in the quota size. Use the Move deleted files to Conflict and Deleted folder feature if you want to restore deleted files. (This feature is enabled by default.)	File conflicts can occur if write permissions are set on the hub and branch servers. In this case, the quota size of the Conflict and Deleted folder should be chosen according to how much data you anticipate having to retrieve due to conflicts. Using the Move deleted files to Conflict and Deleted folder is optional. If you have set up Shadow Copies of Shared folders, you might not need this option, though turning it on will save WAN bandwidth when a user accidentally deletes a file and then immediately restores a previous version.
Namespace	Use to simplify the way clients access distributed data; provides redundancy.	Optional in this scenario; used primarily for client failover and failback purposes.
Referral ordering	Use the Exclude targets outside of client's site referral ordering option and Last among all targets target priority to prevent clients from failing over to servers in other branch sites.	Use default ordering method (lowest cost) if a namespace is used.
Client failback	Optional; enable if you want supported clients to fail back to a local server.	Optional; enable if you want supported branch clients to fail back to the branch server.
Target priority	Use the Exclude targets outside of client's site referral ordering option and Last among all targets target priority to prevent clients	If the branch server is in its own site (as defined in Active Directory), target priority is not needed. Otherwise, use target priority to specify how branch

Setting or Feature	Data Distribution	Data Collection
	from failing over to servers in other branch sites.	and hub servers appear in referrals.
Redundant domain-based namespace servers	Use to increase the availability of domain-based namespaces; place namespace servers in the same sites as clients (or in sites connected by low-cost connections).	Use redundant namespace servers (one in the branch and one in the hub) if you want the namespace to continue to function when the WAN is unavailable.
Root scalability mode (domain-based namespaces only)	Use if you plan to have more than 16 servers host a common namespace.	Use if you have a namespace server in each branch, the namespace servers host a common namespace, and you have more than 16 branches.

Distributed File System Design Process

The following steps outline the general process for designing distributed file systems using DFS Namespaces and DFS Replication. Although the recommendations for each scenario differ, these steps generally apply to each scenario in this guide.

1. Identify data to replicate
2. Make initial namespace decisions
3. Design the replication topology
4. Plan for high availability and business continuity
5. Plan for delegation
6. Design the namespace hierarchy and functionality
7. Design replication schedules and bandwidth throttling
8. Review performance and optimization guidelines
9. Plan for DFS Replication deployment

Plan for Data Publication

The following sections describe the steps for planning for data publication.

Identify Data to Replicate

To begin planning for data publication, identify the servers and folders that contain the data you want to replicate to other servers. You will create a single replication group that contains multiple replicated folders if any of the following statements are true:

- The data to be replicated is located on multiple volumes or cannot be put into one folder tree.
- You want to replicate one or more folders to the full set of replication group members, or to a subset of replication group members.

Using a single replication group allows you to use the same topology for all published data, which simplifies the initial setup and management of replication. Each replicated folder can be replicated to all members or to a subset of members. You will need to create multiple replication groups if:

- Some replicated folders require different topologies.
- Some replicated folders require unique schedules and bandwidth throttling settings.

If the same data already exists on multiple servers that will be part of a replication group, identify the server that contains the most up-to-date version, if necessary. You will choose this server as the primary member when you configure replication. The primary member's data is considered authoritative during initial replication and will win any conflict resolution, even for files that are more up-to-date on the non-primary members.

The following table describes how prestaged files are handled during initial replication.

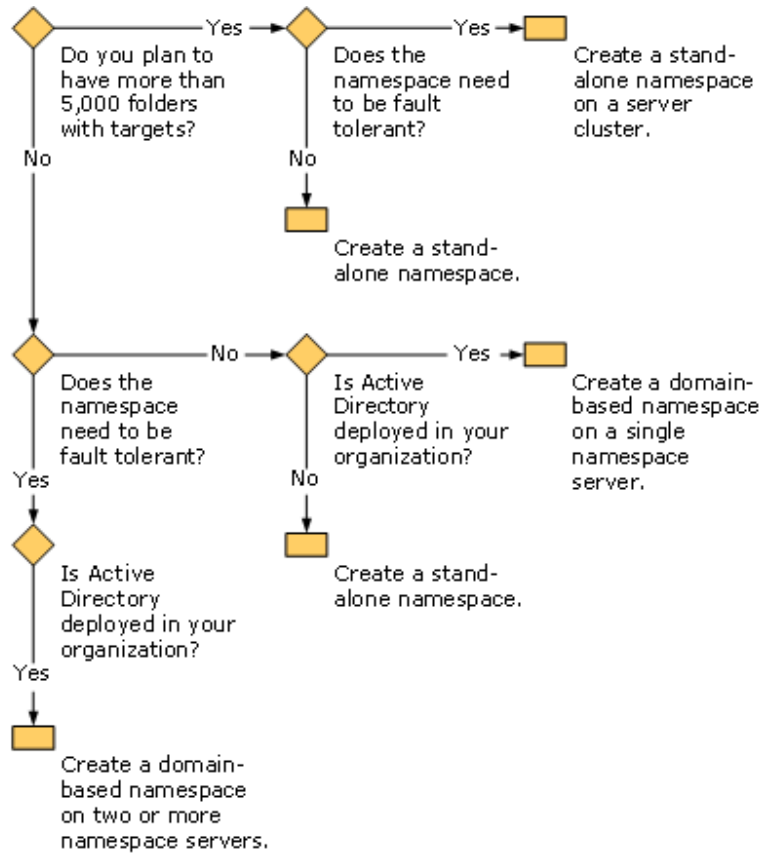
File on Primary Member	File on Non-Primary Member	Result
File A.doc is identical to File A.doc on the non-primary member.	File A.doc is identical to File A.doc on the primary member.	The file is not replicated to the non-primary member. Minimal metadata is replicated, however, to update the DFS Replication database on the non-primary member.
File B.doc is more up-to-date than the version of File B.doc on the non-	File B.doc is outdated compared to the version of File	The primary member's version of File B.doc is considered authoritative. The version of File B.doc on the non-primary member is moved to the Conflict and Deleted folder. File B.doc from the primary member is

File on Primary Member	File on Non-Primary Member	Result
primary member.	B.doc on the primary member.	replicated to the non-primary member. RDC and cross-file RDC can be used to replicate portions of the file to the non-primary member.
File C.doc does not exist on the primary member.	File C.doc exists on the non-primary member.	File C.doc on the non-primary member will be moved to the member's Preexisting folder at the end of initial replication.
File D.doc is outdated compared to the version of File D.doc on the non-primary member.	File D.doc is more up-to-date than the version of File D.doc on the primary member.	The primary member's version of File D.doc is considered authoritative. The version of File D.doc on the non-primary member is moved to the Conflict and Deleted folder. File D.doc from the primary member is replicated to the non-primary member. RDC and cross-file RDC can be used to replicate portions of the file to the non-primary member.
File E.doc does not exist on the primary member.	File E.doc is created on the non-primary member while initial replication is taking place.	File E.doc is replicated to the primary member after initial replication completes.
File G.doc is identical to File G.doc on the non-primary member.	File G.doc is deleted on the non-primary member while initial replication is taking place.	If File G.doc from the primary has not replicated to the non-primary member before the delete occurs, the delete does not replicate. Otherwise, the delete replicates because the delete occurs on the primary member's version.

Guidelines for managing preexisting files are described in "Plan for DFS Replication Deployment" later in this guide.

Make Initial Namespace Decisions

The data publication scenario works well when replicated folders are published in a namespace. Before you design a namespace, you need to answer some basic questions. The following flowchart will help you decide the type of namespace to create, based on your responses to the questions in the sections that follow.



What type of namespace do you want to create?

The size of the namespace can help you determine the type of namespace to choose. Essentially, the size of a namespace is based on the number of namespace folders that each correspond to a target. Typically you will create a namespace folder for each replicated folder, so the number of replicated folders you plan to create can help you decide the size and type of your namespace. If you plan to have fewer than 5,000 namespace folders in a namespace, we recommend using a domain-based namespace if you have deployed Active Directory. If you will create more than 5,000 namespace folders in a namespace, or Active Directory is not deployed in your organization, create a stand-alone namespace, or create multiple domain-based namespaces, each with fewer than 5,000 folders with targets. For more information about these size recommendations, see "DFS Namespaces size recommendations" later in this guide.

Does the namespace need to be fault-tolerant?

A namespace hosted on a single, non-clustered server is not accessible if the server fails. Therefore, you must decide whether you want to make the namespace fault-tolerant, which prevents the situation in which the target servers are up and running but users cannot access them by using the namespace. Each type of namespace (stand-alone or domain-based) is made fault-tolerant in a different way.

- To make a stand-alone namespace fault-tolerant, you create it on a server cluster using the Cluster Administrator snap-in.
- To make a domain-based namespace fault-tolerant, you need at least two namespace servers and two domain controllers. (A client that attempts to access a domain-based namespace will first contact a domain controller for a referral. If a domain controller is not available, the client cannot access the namespace.) If you plan to host folder targets on the same server cluster where the stand-alone namespace is created, you will not be able to use DFS Replication on these folder targets because DFS Replication does not support server clusters.

Is it OK for all branch clients to access a server in a central location to receive referrals?

Decide whether it is acceptable for branch clients to access a domain controller (to request domain-based root referrals) or a namespace server (to request stand-alone root referrals and folder referrals) in a site other than the client's site, such as the hub site. Although clients will cache these referrals for a configurable amount of time, the referrals are purged if the client is rebooted, the referral expires, or the client's referral cache is flushed. The client must then contact the namespace server again for a referral and possibly a domain controller as well if the namespace is a domain-based namespace. (For more information about the referral process, see the DFS Technical Reference on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=36988>).

If you do not want clients to request referrals from a remote domain controller or namespace server, you must place domain controllers and namespace servers in each branch office as follows:

- For domain-based namespaces, place a domain controller and namespace server in the branch office.

- For stand-alone namespaces, place a namespace server in the branch office. For both cases, the namespace server can be the same server as an existing domain controller or file server in the client's site, so it might not be necessary to deploy new hardware. Placing these servers in the client's site will also ensure that the namespace is available if the WAN connection fails. If you don't have a namespace server (and a domain controller if the namespace is domain-based) in the client's site, the client cannot access the namespace if the WAN is down. The tradeoff for putting these servers in the client's site is that they must be monitored to ensure that they are up and running and that the namespace and domain controller roles are healthy.

Design the Replication Topology

The replication topology is a framework of replication paths between members. A bi-directional replication path consists of two one-way connections that replicate data in the opposite direction. For ease in recovery, we recommend always using two one-way connections (as opposed to creating a single one-way connection) between each member, and we recommend using shared folder permissions to prevent changes from being made on branch servers.

To publish data, you will likely use a hub-and-spoke topology, where one or more hub servers are located in data centers, and servers in branch offices will connect to one or more hub servers. To prevent the hub servers from becoming overloaded, we recommend that fewer than 100 spoke members replicate with the hub server at any given time. If you need more than 100 spoke members to replicate with a hub server, set up a staggered replication schedule to balance the replication load of the hub server.

For recommendations regarding the number of connections per member, see "DFS Replication limits" later in this guide.

Plan for Backups

Backups are essential for a highly available distributed file system, because the ultimate recovery method is to restore from backup. Therefore, you must make plans to:

- Regularly back up the namespace. This involves exporting the namespace configuration using the Windows Support Tool Dfsutil.exe. The recovery process

involves creating a new namespace root and then importing the namespace configuration using Dfsutil.exe.

- Regularly back up the replicated data. Because data originates from the hub servers in this scenario, you can perform the backups at the hub site.
- Regularly back up Active Directory. Configuration information for DFS Replication is stored in Active Directory. Therefore, this information will be backed up as part of your regular Active Directory backup process.
- Create an inventory of DFS Replication settings. You can use the Dfsradmin.exe command-line tool to create a list of all replication groups, replicated folders, and their respective attributes.

Plan for Delegation

The permissions required to create and manage namespaces and replication groups vary according to the type of task. Although in some cases membership in the Domain Admins group is required for these tasks by default, it is possible to delegate the ability to perform almost every task associated with namespaces and DFS Replication. Therefore, it is important to determine who will need to perform these tasks so that a member of the Domain Admins group can delegate permissions as appropriate.

Specifically, you will need to determine who will perform the following tasks:

- Create stand-alone namespaces
- Create domain-based namespaces
- Manage existing namespaces
- Create replication groups and enable replication on folder targets in a namespace
- Manage replication groups

For more information about delegating the ability to create and manage namespaces and replication groups, see "DFS Replication security requirements and delegation" and "DFS Namespaces security requirements and delegation" later in this guide.

Design the Namespace

The following sections will help you design the namespace hierarchy, choose the referral ordering method and target priority, and configure client failback.

Namespace hierarchy

To design the namespace hierarchy, you choose the name of the namespace (also called the root name), the names of the folders that will appear beneath the root, and the hierarchy of the folders. The root and folder names should reflect not only your organization's needs but also the type of data you plan to distribute. The following basic guidelines will help you choose namespace and folder names:

- A namespace name is the point beyond the server name or domain name that is at the top of the hierarchy of the logical namespace. Standardized and meaningful names at this level are very important, especially if you have more than one namespace in a domain, because the namespace name is where users enter the namespace.
- The folder names and hierarchy of a namespace must be as clear as possible to the users so that they do not follow the wrong path and have to backtrack. To minimize the number of times that users traverse a wrong path and avoid the delay that users can experience while a connection is set up with an undesired target, develop a meaningful naming scheme for folders in the namespace.
- The namespace does not have to map to the logical organization of files on the file systems; instead, the namespace should map to your organization's business needs.
- The namespace should be independent of geography. For example, creating a namespace path such as \\Contoso.com\Washington\Applications does not make sense even if the Washington users see only a subset of applications. Disable the membership of replicated folders for given servers to control this for publication.

If you plan to create a namespace with a large number of folders, the naming scheme and hierarchy is especially important so users do not have to scan a long list of folders to find the folder they are looking for. Using folders without targets (essentially namespace subfolders) will help you build a deeper hierarchy so that users can make a choice from a small number of top-level folders.

To flatten the namespace and reduce the number of folders that the user sees when browsing the namespace, consider using access-based folder enumeration, first introduced in Windows Server 2003 SP1, to hide folders that

the user does not have access to. For more information about this feature, see the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55319>).

Referral ordering and target priority

When a client computer attempts to access a namespace, a domain controller or namespace server provides a referral to the client. The referral contains a list of target servers that are sorted according to the currently configured ordering method and target priority. When a client accesses the namespace root or a folder in the namespace, the client attempts to access the first target at the top of the referral; the client moves to the next target if the prior target was not available.

The lowest cost ordering method is selected by default. In this method, targets are ordered as follows (assuming no target priority settings override the default behavior for this method):

1. Targets in the same Active Directory site as the client are listed in random order at the top of the referral.
2. Next, targets outside of the client's site are listed in order of lowest cost to highest cost. Referrals with the same cost are grouped together and within each group the targets are listed in random order.

If you want to prevent branch clients from failing over to a branch server at a different branch site, select the **Exclude targets outside of the client site** ordering method for each folder with targets, and then set target priority on each hub server's folder target by selecting the **Last among all targets** target priority. The result of selecting these two options is as follows:

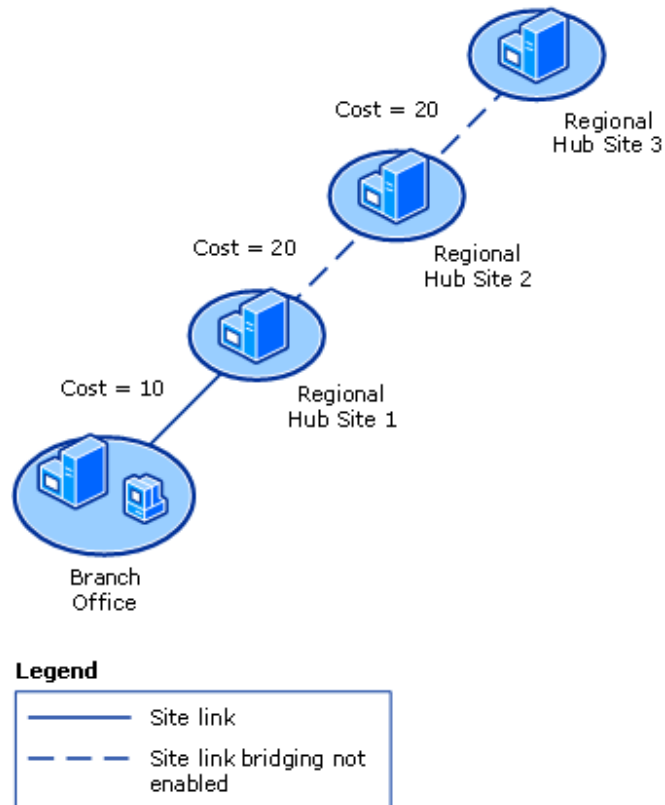
- The **Exclude targets outside of the client site** setting ensures that only targets within the client's site will be included in referrals.
- The **Last among all targets** setting overrides the referral ordering method by including the hub server in the referral, even if the hub server is not in the client's site. (If multiple hub servers are used as folder targets for a given folder, those hub servers will appear last in the referral and be sorted in order of lowest cost after the other targets.)

If you plan to use lowest cost referral ordering and target priority, be aware that domain controllers and namespace servers must run the operating systems described in "DFS Namespaces Requirements" earlier in this guide. If domain

controllers or namespace servers are running Windows 2000 Server, they cannot provide referrals based on cost or priority. Referrals from these servers will use random referral ordering as follows (assuming no target priority settings override the default behavior for this method):

1. Targets in the same site as the client are listed in random order at the top of the referral.
2. Next, targets outside of the client's site are listed in random order. If no same-site target servers are available, the client computer is referred to a random target server no matter how expensive the connection is or how distant the target is.

The lowest cost ordering method works properly for all targets only if the **Bridge all site links** option in Active Directory is enabled. (This option, as well as site link costs, are available in the Active Directory Sites and Services snap-in.) An Intersite Topology Generator that is running Windows Server 2003 relies on the **Bridge all site links** option being enabled to generate the intersite cost matrix that the Distributed File System service requires for its site-costing functionality. If this option is turned off, the Distributed File System service only computes cost for sites that have a direct site link from the branch location to the other sites. All sites that do not contain a direct site link will have the maximum possible cost. For example, assume that the topology between a branch sites and three regional data centers is configured as shown in the following figure:



In this figure, there is a site link between the branch site and the regional data center site 1. No site link is configured between the branch site and the other regional data centers. When a client in the branch site receives a referral, the targets are ordered as follows:

1. The server in the branch site. (Because the server is in the same site as the client, its cost is 0.)
2. The server in regional data center site 1. (This server is listed second because a site link exists between this data center and the branch office.)
3. The next 2 data centers, in random order. (These servers are listed in random order because the Distributed File System service cannot determine their site costs.)

If the **Bridge all site links** option is enabled, the servers in a referral are listed in the following order:

1. The server in the branch site.
2. The server in regional data center site 1. (Cost = 10)
3. The server in regional data center site 2. (Cost = 30)
4. The server in regional data center site 3. (Cost = 50)

Client failback

Client failover in DFS Namespaces is the process in which clients attempt to access another server in a referral after one of the servers fails or is removed from the namespace. This behavior can be undesirable, though, if a client fails over to a hub server and continues to access the hub server even after the branch server is restored. If you want clients to fail back to a preferred local server when it is restored, plan to enable the **Clients fail back to preferred targets** option for the root. Folders with targets will also use failback if this option is selected for the root.

Design Replication Schedules and Bandwidth Throttling

The replication schedule determines the days and times at which replication occurs. You can configure the schedule in 15-minute intervals on a 7-day schedule. When designing a replication schedule, you can choose between two types of schedules:

- Replication group schedule. This schedule applies to all connections in the replication group except connections that have a custom connection schedule.
- Custom connection schedule. This is a unique schedule that is applied to an individual connection.

When using DFS Replication for data distribution, you will likely set up replication windows so that replication occurs at night or other periods of low network activity, as appropriate for each branch office. If you plan to schedule replication to occur during a replication window, we recommend that the replication window, combined with the bandwidth throttling setting, allows DFS Replication to replicate a higher than average or expected peak of the number of new or changed files within the window. The amount of data that can replicate during the window will depend on the replication throughput, which is determined by a number of factors:

- The number and size of new and changed files
- The bandwidth throttling settings
- The speed of the network
- The ability to compress changes using RDC and whether cross-file RDC is used
- The size of the staging folder quota

- The speed of the disk subsystem
- Whether you have optimized the servers by placing the replicated folder and staging folders on separate disks

DFS Replication efficiency using compression will vary based on the type of files. For example, text files compress very well; files that are already compressed, such as Windows Installer .msi files, will not compress as much. RDC is also very effective against modified files, and cross-file RDC works well when files are new but similar to existing files on the receiving member.

To determine the replication rate, perform testing in a lab environment that resembles your production environment. Use the built-in diagnostic reporting (in the DFS Management snap-in) or Dfsrdiag.exe to watch the replication backlog; the backlog count should go to zero before the replication window closes. If a backlog remains after the replication window closes, then some servers will have stale or missing data.

If the amount of data changes exceeds what DFS Replication can replicate in a given period of time, you need to change one of these factors. You might also want to consider allowing a longer replication window, such as over a weekend, for the rare cases for when a large set of data is modified during the week. Creating an extended replication window during the weekend will allow the backlogged files accumulated during the week to be replicated.

Because members can exist in different time zones, it is also important to consider how the schedule is affected by time zones, and whether daylight savings time is in effect. Replication is always initiated by the receiving member; therefore, the schedule reflects the time at which a receiving member initiates replication with a sending member. The receiving members can interpret the schedule in one of the following ways:

- Universal Coordinated Time (UTC). This option causes the receiving member to treat the schedule as an absolute clock. For example, a schedule that begins at 0800 UTC is the same for any location, regardless of time zone or whether daylight savings time is in effect for a receiving member. For example, assume that you set replication to begin at 0800 UTC. A receiving member in Eastern Standard Time would begin replicating at 3:00 A.M. local time (UTC - 5), and a receiving member in Rome would begin replicating at 9:00 A.M. local time (UTC

+ 1). Note that the UTC offset shifts when daylight savings time is in effect for a particular location.

- Local time of receiving member. This option causes the receiving member to use its local time to start and stop replication. Local time is determined by the time zone and daylight savings time status of the receiving member. For example, a schedule that begins at 8:00 A.M. will cause every receiving member to begin replicating when the local time is 8:00 A.M. Note that daylight savings time does not cause the schedule to shift. If replication starts at 9 A.M. before daylight savings time, replication will still start at 9 A.M. when daylight savings time is in effect.

When the schedule is open, replication occurs as files are changed, created, or deleted. When the schedule closes, replication stops, regardless of whether all changed or new files have replicated.

Review Performance and Optimization Guidelines

The following sections discuss performance and optimization guidelines for data publication scenarios.

Optimize the staging folder quota and replication throughput

In a typical data publication scenario, a large amount of data, such as one or more software programs, originates at a hub site and is replicated out to branch servers, typically on a staggered schedule. Because the initial amount of data to be replicated is large, and because the data will be replicated to many servers, it is important to consider how replication throughput is impacted by the size of the staging folder quota on the hub servers. The quota should be based on the following guidelines:

Size the quota to avoid restaging files

DFS Replication uses staging folders to act as caches for new and changed files to be replicated from sending members to receiving members. The sending member begins staging a file when it receives a request from the receiving member. The process involves reading the file from the replicated folder and building a compressed representation of the file in the staging folder. This is the staged file. After being constructed, the staged file is sent to the receiving member; if remote differential compression [RDC] is used, only a fraction of the

staging file might be replicated. The receiving member downloads the data and builds the file in its staging folder. After the file has completed downloading on the receiving member, DFS Replication decompresses the file and installs it into the replicated folder.

Each replicated folder has its own staging folder, which by default is located under the local path of the replicated folder in the DfsrPrivate\Staging folder. The default size of each staging folder is 4,096 MB. This is not a hard limit, however. It is only a quota that is used to govern cleanup and excessive usage based on high and low watermarks (90 percent and 60 percent of staging folder size, respectively). For example, when the staging folder reaches 90 percent of the configured quota, the oldest staged files are purged until the staging folder reaches 60 percent of the configured quota.

For best performance, size the staging folder quota on the hub servers to avoid restaging files. The reason for this is as follows: if a file needs to be replicated to one branch server, and later that file needs to be replicated to another branch server (due to staggered replication schedules), one of two results can occur:

- If the staging quota is large, the file is likely already staged and does not need to be restaged. This will reduce the CPU and disk I/O needed to replicate the file to subsequent branch servers.
- If the staging quota is low, the staging file is more likely to have been purged as described earlier, and the file will need to be restaged before it is replicated. This will increase the CPU and disk I/O needed to replicate the file to subsequent branch servers.

If you plan to use staggered replication schedules, it is important to size the staging folder quota so that it is large enough to store the amount of data that will be replicated during the replication window. For example, if you plan to drop 2 gigabytes (GB) of files into the replicated folder, and you want that data to replicate within the replication window (assuming the window is long enough), set the staging quota size to at least 2 GB. That way, the full 2 GB of data will still be in the staging folder when the next replication window opens on the staggered schedule. (Note that DFS Replication compresses the staging files, so a 2 GB quota might not be necessary.)

RDC also performs better when files are kept in the staging folder. The reason for this performance increase is as follows: When a file is changed and subsequently staged in the staging folder, RDC logically "breaks" the file into portions and uses an algorithm to generate a checksum value for each portion. By comparing the checksums on the sending and receiving members, RDC identifies the mismatched checksums, which indicate that the portion has changed, and then replicates only the changed portions.

These checksums are stored as an alternate data stream on the staging file. As long as the file is kept in the staging folder, the checksums do not need to be regenerated if the file is replicated again. Generating the checksum value does require CPU overhead, which is why it is good to minimize checksum regeneration by keeping staged files. However, after the file is replicated and installed into in the replicated folder on the receiving member, the checksum is removed from the file.

Size the quota to avoid having the staging folder reach or exceed the configured quota

If large files are being replicated, make sure the staging folder quota on the hub server is approximately ten times larger than the largest nine files. To understand this guideline, consider the following functions of DFS Replication:

- The staging folder is purged when it reaches 90 percent of the configured quota.
- If the staging folder size is below 100 percent of the configured quota, DFS Replication will typically replicate nine files at a time. (The 9 files consist of 5 serving threads and 4 download threads, assuming the server is participating in both data publication and data collection.)

A decrease in replication throughput can occur if the staging folder reaches 100 percent or more of its configured quota, such as when a large file needs to be replicated. In this case, the file will be staged (along with other previously staged files), which causes two events to occur:

- Staging folder cleanup is triggered, because the staging folder has exceeded 90 percent of its configured quota.

- DFS Replication stops replicating nine files at a time and instead replicates one file at a time until the staging folder is less than 90 percent of the configured quota.

To avoid this potential decrease in replication throughput, ensure that each of the nine largest files will consume no more than 10 percent of the staging folder quota.

Optimize the Conflict and Deleted folder quota

Although conflicts are unlikely in a data publication scenario, you might want to use the Conflict and Deleted folder to store accidental deletions. The benefit of doing so is that when you restore the files on one member, RDC will be used on the receiving members to reconstruct the deleted files and put them back into the replicated folder without replicating them in full over the WAN, thus saving network bandwidth.

When sizing the Conflict and Deleted folder quota for deleted files, consider how much data might be accidentally deleted; in a publication scenario, this could be a large amount of data if an entire tree of program files, for example, is accidentally deleted.

Optimize the amount of time that clients cache referrals

For clients that are not running Windows XP with SP2 or Windows Server 2003 with SP1, the cache duration for a referral determines the earliest time that a client will request a new referral, but only if the existing referral expires before it is accessed again. Clients that use a cached referral will renew the Time to Live of the referral and thus use the referral indefinitely until the client's referral cache is cleared or the client is restarted.

This behavior has changed for clients running Windows XP with SP2 or Windows Server 2003 with SP1. Specifically, the Time to Live value is not reset each time a client accesses a target using a cached referral. Instead, the referral expires after the Time to Live value lapses. This change has several effects:

- Clients running Windows XP with SP2 or Windows Server 2003 with SP1 will request referrals more frequently than other clients, which can cause moderately increased load on the domain-based namespace servers and domain controllers.

- Because they request new referrals more frequently, clients running Windows XP with SP2 or Windows Server 2003 with SP1 will discover namespace updates more quickly than other clients.

By default, clients cache referrals for namespace roots for 300 seconds (5 minutes) and referrals for folders with targets for 1800 seconds (30 minutes).

In branch office environments, adjust these values using the following guidelines:

- If a namespace server and domain controller exists in the branch office, use the default cache duration for both namespace roots and folders with targets. You do not need to adjust these values because the clients in the branch do not need to access the WAN to request referrals.
- If a namespace server or domain controller does not exist in the branch office, and clients must access the WAN to request a referral, increase the cache duration. Doing so serves two purposes: clients will request referrals less frequently, and clients can continue to access previously visited portions of the namespace even when the WAN is down (assuming the targets servers are accessible). When increasing this value, however, keep in mind that the cache duration also affects when client failback is triggered. The longer the cache value, the longer clients will continue to access an out-of-site server if a same-site server is later restored.

Plan for DFS Replication Deployment

In the section "Identify Data to Replicate," you identified the servers that contain the data you want to replicate, the servers to participate in replication, the servers to be prestaged, and the server that contains the most up-to-date version of the data. Ensure that these servers will run Windows Server 2003 R2 and that the DFS Replication service is installed on these servers prior to your deployment. You must also ensure that enough hard disk is available on each member. You need to account for the amount of data in each replicated folder, potential growth of the replicated data, and the quota sizes of the staging and Conflict and Deleted folders that are created for each replicated folder.

If time is not a concern, you can enable replication using highly throttled bandwidth and let the data replicate over a number of days or as long as necessary for initial replication to complete.

After initial replication completes, plan to review the Preexisting folder on each non-primary member so that you can delete the files to reclaim disk space (these files are never purged by DFS Replication) or move the files back into the replicated folder tree, if desired. You should also review the Conflict and Deleted folder manifest on non-primary members to identify any conflicting files (files with same name but different data as compared to the primary member) that you want to retrieve. It is important to review the conflicted or deleted files because they will be purged when the high watermark (90 percent of staging folder quota) is reached. (You can view a log of conflict files, as well as their original file names, by viewing the ConflictandDeletedManifest.xml file in the DfsrPrivate folder under the local path of the replicated folder.)

If you plan to publish replicated folders in a namespace, we recommend that you disable referrals to the non-primary members' folder targets until initial replication is complete. This will prevent users from accessing the servers until the data has completely replicated, at which point you can enable referrals to those folder targets.

Plan for Data Collection

The following sections describe the steps for planning for data collection.

Identify Data to Replicate

To begin planning for data collection, identify the servers in branch offices that contain the data that you want to replicate to a server in the hub site. Typically you will set up a replication group for each hub server/branch server pair.

Setting up individual replication groups (for each branch office) has the following benefits over setting up a single replication group to contain all branch servers and all branch-related replicated folders:

- You do not need to disable memberships to prevent one branch server's data from being replicated to another branch server. Replication performance and DFS Management snap-in performance are also improved when you do not have a large number of disabled memberships.
- You can choose a replication schedule and bandwidth throttling settings best suited for each branch office. For example, time zone differences can come into play if the branch offices are geographically distributed, so the schedule for one

branch office might not work for another. By using individual replication groups for each branch server/hub server pair, you can adjust the replication window as appropriate for each branch office.

- Because delegation is done at the replication group level, you have better control in delegating management permissions.

It is also important to consider how NTFS permissions work in the data collection scenario. Specifically, the NTFS permissions for a replicated folder on the branch server will be applied to the same replicated folder on the hub server. Because the type of data collected is typically user data, you must ensure that users can change files from only one server to avoid change conflicts. Publishing the replicated folders in a namespace can ensure that branch clients access the replicated folder in their own site first and fail over to the hub server only if the branch server is unavailable. If you do not plan to use a namespace, ensure that shared folder permissions on the hub server prevent users from changing files there.

Because files are replicated only after they are closed, using DFS Replication for data collection is not advised for databases or any other types of files that are held open for long periods of time. If the files to be replicated include .pst files, we recommend that you configure Outlook® to periodically to release locks on .pst files. Depending on the version of Outlook, you might also need to install an Outlook hotfix. If Outlook is not configured properly, sharing violations will occur and the .pst files will not be replicated because they are held open. The recommended procedures are:

- Outlook 2000. Refer to Knowledge Base article 222328 for information about decreasing the timeout period. This article is available in the Knowledge Base on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55321>).
- Outlook 2003. Contact Microsoft Product Support Services to obtain the post-SP1 hotfix package described in Knowledge Base article 839647, available on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55324>). After you install the fix, follow the same process as for Outlook 2000 except use the following registry key:

HKEY_CURRENT_USER\Software\Microsoft\Office\11.0\Outlook\PST key

If there are types of files that you do not want to back up and therefore want to exclude from replication, you can set up replication filters.

Make Initial Namespace Decisions

Although a namespace is not required for this scenario, using a namespace can provide the following benefits:

- Users can navigate the logical namespace without having to know the physical server names or shared folders hosting the data. Because of this, you can physically move data to another server without having to reconfigure applications and shortcuts and without having to re-educate users about where they can find their data.
- A namespace can provide increased data availability. If the branch server fails, clients can fail over to the hub server. However, if you use a namespace for failover purposes, you need to make sure that the hub server's version of the data is kept relatively up-to-date with respect to the branch server. If you plan to replicate data only at night (when backups are performed), you must decide whether it is acceptable for users to access the data from the previous day on the hub server.

If you do not need branch clients to fail over to the hub server if the branch server fails, you do not need to deploy a namespace. If you need failover and plan to deploy a namespace, you need to consider the following questions.

How many namespaces do you want to create?

For data collection, we recommend that you set up one namespace for each branch server/hub server pair if you do not want branch clients to browse to other branch servers using the namespace. If data collaboration is desired across branches, you can add the branch server targets and hub server targets to the same namespace.

For redundancy, the branch server and hub server can both host the namespace. Namespaces can be hosted on member servers and domain controllers, so you do not need to deploy additional hardware to host the namespace if a server already exists in the branch office. If namespace redundancy is not required, you can host the namespace on the branch server only.

If branch office servers will host more than one domain-based namespace, you must install the hotfix described in article 903651 in the Knowledge Base on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55325>) to enable servers running Windows Server 2003, Standard Edition, to host more than one domain-based namespace.

What type of namespace do you want to create?

In the data collection scenario, the namespace will likely be small if it is created for a specific branch office. Therefore, we recommend that you create a domain-based namespace if the following conditions are true:

- The branch server has fewer than 5,000 replicated folders. Each replicated folder will correlate to a folder in the namespace
- You have Active Directory deployed in your organization.

If you choose to create a domain-based namespace, you need to determine whether it is acceptable for clients to request referrals from namespace servers outside of the branch site and whether you want to make the namespace fault-tolerant. Both of these issues are covered in the following sections.

Does the namespace need to be fault-tolerant?

A namespace hosted on a single, non-clustered server is not accessible if the server fails. Therefore, you must decide whether you want to make the namespace fault-tolerant, which prevents the situation in which the target servers are up and running but users cannot access them by using the namespace.

To make a domain-based namespace fault-tolerant, you need at least two namespace servers and two domain controllers. (A client that attempts to access a domain-based namespace will first contact a domain controller for a referral. If a domain controller is not available, the client cannot access the namespace.)

If it acceptable for users to access the target servers directly instead of by using the namespace, you do not need a fault-tolerant namespace.

Is it OK for all clients to access a server in a central location to receive referrals?

Decide whether it is acceptable for clients to access a domain controller (to request domain-based root referrals) or a namespace server (to request folder referrals) in a site other than the client's site, such as the hub site. Although

clients will cache these referrals for a configurable amount of time, the referrals are purged if the client is rebooted, the referral expires, or the client's referral cache is flushed. The client must then contact the domain controller or namespace server again for a new referral. If you do not want clients to request referrals from a remote domain controller or namespace server, you must place a domain controller, namespace server, or both in the client's site. (The namespace server can be the same server as an existing domain controller or file server in the client's site.) Doing so will also ensure that the namespace is available if the WAN connection fails; the tradeoff is that these servers must be monitored to ensure that they are up and running.

Design the Replication Topology

If you use the New Replication Wizard to set up a replication group for each branch server, the wizard sets up two one-way connections between the hub server and the branch server. This means that changes made at the hub server will be replicated to the branch server and vice versa, which could result in file conflicts if users are accessing both servers. Setting shared folder permissions can help prevent conflicts from occurring. Or, if you are using a namespace, you can disable referrals to the hub server so that clients are directed only to the branch server. If the branch server fails, you will need to re-enable the referrals to the hub server.

Because the hub server will be part of multiple replication groups (and will therefore host multiple replicated folders), you must take into account the fact that each replicated folder will have its own staging folder and Conflict and Deleted folder, and each of these folders has its own quota. Therefore, you need to ensure that the hub server has adequate disk space to store files in the staging and Conflict and Deleted folders.

Plan for Backups

Backups are essential for a highly available distributed file system, because the ultimate recovery method is to restore from backup. Therefore, you must make plans to:

- Regularly back up the namespace. This involves exporting the namespace configuration using the Windows Support Tool Dfsutil.exe. The recovery process

involves creating a new namespace root and then importing the namespace configuration using Dfsutil.exe.

- Regularly back up the replicated data. Because data from the branch offices is replicated to the hub servers in this scenario, you can perform the backups at the hub site.
- Regularly back up Active Directory. Configuration information for DFS Replication is stored in Active Directory. Therefore, this information will be backed up as part of your regular Active Directory backup process.
- Create an inventory of DFS Replication settings. You can use the Dfsradmin.exe command-line tool to create a list of all replication groups, replicated folders, and their respective attributes.

Plan for Delegation

The permissions required to create and manage namespaces and replication groups vary according to the type of task. Although in some cases membership in the Domain Admins group is required for these tasks by default, it is possible to delegate the ability to perform almost every task associated with namespaces and DFS Replication. Therefore, it is important to determine who will need to perform these tasks so that a member of the Domain Admins group can delegate permissions as appropriate.

Specifically, you will need to determine who will perform the following tasks:

- Create stand-alone namespaces
- Create domain-based namespaces
- Manage existing namespaces
- Create replication groups and enable replication on folder targets in a namespace
- Manage replication groups

For more information about delegating the ability to create and manage namespaces and replication groups, see "DFS Replication security requirements and delegation" and "DFS Namespaces security requirements and delegation" later in this guide.

Design the Namespace

If you plan to create a namespace for each branch office, the namespace hierarchy can be relatively simple. The name of the namespace root can match the branch office's name or purpose, and each namespace folder can represent a replicated folder available at the branch.

Assuming the branch clients are in same site as the branch server, clients will always connect to the branch server first. This is because the namespace referral will always list same-site servers first (assuming no target priority settings override this behavior). If the branch server is unavailable, the branch clients will fail over to the hub servers. To enhance the namespace functionality, consider setting the following options.

- **Client failback.** If you want clients to fail back from the hub server to the branch server (after the branch server is restored), enable client failback for the namespace root; this setting will be inherited by all folders with targets, which represent replicated folders.
- **Target priority.** If you have two hub servers, the branch client will fail over to either hub server without preference for one or the other. If you prefer clients to fail over to a particular hub server, select the option **Last among all targets** on the non-preferred hub server's root target.

Design Replication Schedules and Bandwidth Throttling

In the data collection scenario, you will typically replicate at night during a replication window. Waiting to replicate changes at night is beneficial because, as changes are made throughout the day, they are not immediately replicated like they would be if the schedule were open 24 hours a day. Instead, the changes are collected (or dampened), and thus the changes are only replicated once when the replication window opens.

To determine the length of the replication window, you will need to analyze the amount of data that would need to be replicated as a result of the day's changes and take into account the time needed to back up the changed files. If the replication window is longer than the amount of time needed to replicate the files, you can adjust the bandwidth so that less bandwidth is used during the window, or you can leave the bandwidth as-is to deal with unexpected surges in replication traffic.

The downside to replicating only at night is that files on the hub server are not up-to-date during the day. If the branch server fails, users will not be able to access the most recent files on the hub server. This can be an issue if you are using a namespace and users transparently fail over to the hub server. If the hub server must be up-to-date, you will need to replicate continuously.

If the hub server will host a number of branch servers' data, consider staggering the replication schedules to reduce the load on the hub server. Staggering might also be necessary if branch offices are located in different time zones. Because replication is always initiated by the receiving member, the schedule reflects the time at which a receiving member initiates replication with a sending member. Therefore, adjust the schedules accordingly to ensure that replication occurs at off-hours for both hub servers and branch servers.

Review Performance and Optimization Guidelines

In the data collection scenario, changes are made throughout the day on branch servers, and those changes are queued in the staging folder for replication at night. Therefore, it is important to size the staging folder quota to accommodate the changes that will be replicated when the replication window opens. Doing so ensures that files remain in the staging folder and are not purged during the day, which can affect replication throughput if the files need to be restaged before they are replicated. Having files in the staging folder also increases RDC performance.

If large files are being replicated, make sure the staging folder quotas on the branch servers are approximately ten times larger than the largest nine files. The reasons behind this recommendation are described in the data publication scenario earlier in this guide.

Plan for DFS Replication Deployment

If branch files are not prestaged on the hub server, initial replication can take a long time to complete. If time is not important but limiting bandwidth used is important, set the bandwidth throttling to a low setting and let the data trickle down from the branch servers to the hub server. To determine whether the hub server has received the initial files (at which point you can begin backing up the data at the hub server), check the backlog size reported in the diagnostic report,

use Dfsrdiag.exe to check backlog, or manually compare the number of files on the hub and branch servers.

If you need initial replication to complete more quickly, you can prestage the data on the hub server by backing up the branch servers' data to media. Ship the media to the hub site and copy the files to the replicated folder on the hub server. Then, when you set up replication, only the changes are replicated from the branch server to the hub server.

For more information about prestaging, see the "Plan for DFS Replication Deployment" section in the data publication scenario earlier in this guide.

Plan for Monitoring

There are two methods for monitoring DFS Replication: the built-in diagnostic reporting (in the DFS Management snap-in) and the DFS Replication MOM Pack. The DFS Replication MOM Pack is designed for monitoring 50 or more servers. The diagnostic report is suitable monitoring up to 50 servers; if used for monitoring more than 25 servers, the diagnostic report might take a long time to generate, and the resulting file can be quite large and slow to open in a Web browser.

Review DFS Replication Requirements

The following sections describe the various requirements of DFS Replication that apply to all scenarios described in this guide.

Active Directory requirements for DFS Replication

- Active Directory must be deployed in the organization.
- The Active Directory schema must be updated to include the new DFS Replication classes and attributes. These schema changes are provided on the second Windows Server 2003 R2 operating system CD. This schema can be applied to domain controllers running Windows 2000 Server, Windows Server 2003, and Windows Server 2003 R2.
- All members of a replication group must be in the same forest. You cannot enable replication across servers in different forests.

Operating system requirements for using and managing DFS Replication

The servers that will participate in DFS Replication must run Windows Server 2003 R2. After you install Windows Server 2003 R2, you must install the DFS Replication Service on each server that will take part in replication, and you must install the DFS Management snap-in on one server to manage replication.

You can also manage DFS Namespaces and DFS Replication from a computer running Windows XP with Service Pack 2 (SP2) by installing the Administration Tools Packs for Windows Server 2003 R2. When this pack is installed, the DFS Management snap-in is available as part of the File Server Management snap-in. For more information about installing this pack, see the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55225>).

DFS Replication compatibility

- File systems. Replicated folders must be stored on NTFS volumes.
- Server clusters. On server clusters, replicated folders should be located in the local storage of a node, because the DFS Replication service is not designed to work in a coordinated way with cluster components, and the service will not fail over to another node.
- Files not replicated. DFS Replication does not replicate the following types of files:
 - NTFS mounted drives within the local path of the replicated folder. (However, the local path of a replicated folder can be at or under a mounted drive.)
 - Files encrypted by using encrypting file system (EFS).
 - Any reparse points except those associated with DFS Namespaces. If a file has a reparse point used for Hierarchical Storage Management (HSM) or Single Instance Store (SIS), DFS Replication replicates the underlying file but not the reparse point.
 - Files on which the temporary attribute has been set.
 - Files with case-sensitive names created by using a UNIX-compatible application and saved to a Network File System (NFS) shared folder. (NTFS supports the Portable Operating System Interface [POSIX] standard, which provides the ability to create two files that have the same name but different capitalization.) For example, if a user creates two files, FILE.doc and File.doc,

on an NFS shared folder, DFS Replication will replicate the first but not the second.

- Quota software. If you plan to use disk quotas, follow these guidelines:
 - Disable any quotas on the DfsrPrivate folder that is automatically created under the local path of each replicated folder. Doing so ensures that disk quotas do not prevent DFS Replication from using the staging folder and Conflict and Deleted folder as configured.
 - Set quotas identically on all members.
 - Disable quotas on the \System Volume Information\DFSR folder that is in the root of the system volume and all volumes that contain replicated folders.
 - Watch for disk full events. Disk full events can be caused by quotas on certain folders even when there is sufficient space on the volume.
 - For specific guidelines for using the quotas in the File Server Resource Manager, see "Interoperability guidelines for File Server Resource Manager" later in this guide.
- Ultrasound and Sonar. The Ultrasound and Sonar monitoring tools designed for File Replication service (FRS) are not compatible with DFS Replication. Use the built-in diagnostic reporting or the DFS Replication MOM pack to monitor DFS Replication.
- Firewalls. DFS Replication might not work across firewalls because it uses the RPC dynamic endpoint mapper. Additionally, configuring DFS Replication using the DFS Management snap-in does not work when a firewall is enabled. You must define a port exception or disable the firewall, or you can use Dfsrdiag.exe to set the static RPC port for DFS Replication. For more information, see the DFS Replication Operations Guide on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55327>).
- SYSVOL. DFS Replication is not supported for SYSVOL replication in Windows Server 2003 R2. Do not attempt to configure DFS Replication on SYSVOL by disabling FRS and setting up a replication group for SYSVOL. Continue to use FRS for SYSVOL replication on domain controllers running Windows Server 2003 R2. FRS and DFS Replication can coexist on the same member server or domain controller.

- Antivirus software. Antivirus software must be compatible with DFS Replication; contact your antivirus software vendor to check for compatibility. (With RDC, excessive replication does not occur due to incompatible virus software, but there is overhead associated with the RDC protocol.)

DFS Replication security requirements and delegation

The following table describes the groups that can perform basic DFS Replication tasks by default and the method for delegating the ability to perform these tasks.

Task	Users or Groups That Can Perform This Task By Default	Delegation Method
Create a replication group or enable DFS Replication on a folder that has folder targets	Domain Admins group in the domain where the replication group will be created.	Right-click the Replication node in the console tree, and then click Delegate Management Permissions .
Administer a replication group	Domain Admins group in the domain where the replication group is configured, or the creator of the replication group.	Right-click the replication group in the console tree, and then click Delegate Management Permissions .
Add a server to a replication group ^{1, 2}	If the server is a member server, the user must be a member of the local Administrators group of the server to add. If the server is a domain controller, the user must be a member of the Domain Admins group in the domain where the server is located.	Add the user to local Administrators group of the member server to add, or add the user to the Domain Admins group of the domain controller to add.

¹Assumes that the user has been delegated the ability to administer the replication group.

² The server to be added must be online.

If you plan to delegate the ability to create and administer replication groups, note the following two important considerations:

- If you delegate to a user or group the ability to create replication groups, and you later remove the user or group from the delegation list, there is no change to the security settings on existing replication groups.
- If you delegate to a user or group the ability to administer a specific replication group, and you later remove the user or group from the delegation list, there is no change to the security settings on any existing configuration data. For example, if the user who is being removed had created a connection in the replication group, then the user would still have permissions to edit that connection because the user is the owner of the Active Directory object that contains the configuration information for the connection.

DFS Replication scalability limits

We recommend that your deployments approach but not exceed the following tested limits:

- Each server can be a member of up to 256 replication groups.
- Each replication group can contain up to 256 replicated folders.
- Each server can have up to 256 connections (for example, 128 incoming connections and 128 outgoing connections).
- On each server, the number of replication groups multiplied by the number of replicated folders multiplied by the number of simultaneously replicating connections must be kept to 1024 or fewer. (If the replication schedule is staggered, you do not need to count the connections that are not replicating due to a closed schedule.)
- A replication group can contain up to 256 members.
- A volume can contain up to 8 million replicated files, and a server can contain up to 1 terabyte of replicated files. These are tested numbers and are recommended guidelines for performance and scalability reasons.

Review additional guidelines and considerations for DFS Replication

The following sections contain guidelines and considerations that apply to all DFS Replication scenarios.

Additional information about DFS Replication staging folders

Staging folders are an important part of the replication process. You should take the following considerations into account when sizing the staging folder quotas for your deployment.

- Staging cleanup can be triggered before the high watermarks are hit if a disk full condition is detected on a volume while the DFS Replication service is performing certain replication activities. The special cleanup will attempt to trim in half all staging folders and all Conflict and Deleted folders on the volume that encountered the disk full condition.
- Staging folders are managed on a per-replicated folder basis. If a server has many replicated folders, especially if there are several located on the same volume, staged files in the staging folders might consume a large portion of the volume and lead to disk full conditions that end users might encounter. To simplify staging management, we recommend that you dedicate a separate volume for staging folders. An alternative is to configure the staging folder path to be the same for all replicated folders on a given volume and use a quota system to configure a quota over that folder. For example, for all replicated folders on volume D:\, configure D:\Staging as the staging folder, and set a quota of 30 GB on the D:\Staging folder.
- Staging cleanup is not necessarily bad if there are only two members and they are replicating with each other. In addition, during initial replication, frequent staging folder cleanup is expected. However, if a hub server is replicating to many partners, frequent staging cleanups are undesirable and you should consider increasing the size of the staging folder quota to avoid cleanups.

Replicating the root of the volume

The DFS Replication service supports configuring a volume root, such as C:\, as the local path of a replicated folder, but this configuration is not recommended for the following reasons:

- DFS Replication replicates metadata (including ACLs and attributes) that are set on the local path of the replicated folder. Volume roots are special because they always implicitly have both the hidden and system attribute bits set. If replication partners are configured to have the corresponding local path of the replicated folder to be some place other than a volume root, that replicated folder will have system and hidden attributes set.

- System folders are typically placed in the volume root. DFS Replication is designed to ignore these folders, but there might be interoperability issues with third-party applications.

Choosing the replication group domain in cross-domain configurations

DFS Replication supports having replicated folders on members in different domains within the same forest. When you set up DFS Replication in a multiple-domain environment, you must choose a domain in which to hold the Global Settings objects in Active Directory. Ideally you want to place the replication group in a domain that minimizes the total site link cost for each member server to the closest domain controller in the replication group's domain. Doing so reduces the cost when the DFS Replication service on each member polls Active Directory as follows:

- The DFS Replication service polls Active Directory at regular intervals (once per hour by default) to get the current configuration. The service will access the LocalSettings object, which is under the member server's computer object in Active Directory, and the objects under the GlobalSettings object.
- The DFS Replication service will also perform lightweight polling (every 5 minutes) by polling only the computer's LocalSettings object. Lightweight polling can be disabled.
- Any change detected by a lightweight poll will trigger a full poll.

Choosing between multiple replication groups, multiple replicated folders, or multiple folders under a replicated folder

DFS Replication offers flexibility in replicating folders among servers. For example, if you have three folders to replicate, you can:

- Create one replication group, and replicated folder, and put the three folders into the replicated folder.
- Create one replication group and three replicated folders that correspond to each of the folders you want to replicate.
- Create three replication groups, each with a single replicated folder that corresponds to one of the folders you want to replicate.

To help you choose among these options, consider the following issues:

- If you have a fast network, consider creating a single replicated folder with multiple subfolders instead of creating a replicated folder for each folder to

replicate. Using multiple replicated folders increases the number of concurrent downloads, resulting in higher throughput, but the trade off is that this configuration can potentially cause heavy disk I/O usage for staging file creation, staging folder cleanup, and so forth. Creating a single replicated folder with many subfolders is a way to throttle the amount of data replicated concurrently and thus minimize disk I/O.

- Replicated folders can be disabled and enabled, allowing for selective memberships.
- Replicated folders have dedicated staging folders and Conflict and Deleted folders. Management of these folders increases as the number of replicated folders increases.
- Delegation, topology, and replication schedules can be made on a per-replication group basis.
- Performance wise, if there is the same number of replicated folders, it should not matter if they are spread across one replication group or multiple replication groups.
- There can be performance benefits to spreading the replicated folders across multiple volumes. To do this, you must use multiple replicated folders, not a single replicated folder.
- Essentially, you should consider management first and performance second when deciding the number of replication groups and replicated folders to create. Creating one replication group with multiple replicated folders might be the best balance. If the folders are relatively small, using a single replicated folder (with multiple subfolders) is fine.
- For high-priority content, creating a separate replication group will provide the flexibility of detailed control of scheduling as well as the option to force replication on a specific connection.

Interoperability guidelines for File Server Resource Manager

If you plan to use DFS Replication on volumes that are monitored using File Server Resource Manager, note the following guidelines:

- If you are using auto quotas on the local path of the replicated folder, disable or delete the quota that is automatically created for DfsrPrivate and its

subfolders. (You might need to right-click the **Quotas** node and then click **Refresh** to cause the DfsrPrivate quota to appear in the list.)

- Do not enable quotas on the local path of the replicated folder, because doing so will also count the DfsrPrivate subfolder. Because quotas cannot be set to ignore a particular subfolder tree, you must enable quotas on individual folders within the local path of the replicated folder.

Review DFS Namespaces Requirements

Operating system and server requirements for DFS Namespaces

- Servers where namespace management tasks are performed using the DFS Management snap-in must run Windows Server 2003 R2 or Windows XP (if the Administration Tools Packs for Windows Server 2003 R2 is installed). For more information about installing this pack, see the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=55225>).
- To support new namespace features (client failback, target priority, and enhanced delegation), all servers that host namespaces must run Windows Server 2003 with SP1 or Windows Server 2003 R2.
- To support new namespace features, all domain controllers must run Windows Server 2003 with SP1 or Windows Server 2003 R2.
- Namespaces must be created on NTFS volumes.

It is possible to use DFS Namespaces when domain controllers and namespace servers run a mix of Windows Server 2003 R2, Windows Server 2003 with SP1, Windows Server 2003 without SP1, and Windows 2000 Server, but some functionality is disabled or available inconsistently, depending on the operating systems on the servers. Some examples of mixed-mode behavior are as follows:

- If domain controllers or namespace servers are running Windows Server 2003 without SP1, they cannot provide referrals that support target priority or client failback.
- If domain controllers or namespace servers are running Windows 2000 Server, they cannot provide referrals that support target server priority or client failback, nor can they order targets by lowest cost in referrals. Additional configuration is required to enable these namespace servers and domain controllers to detect the site of each target server in the namespace. For details, see the answer to the question "What are the issues to consider when I use

multiple servers running Windows 2000 Server and Windows Server 2003 to host a domain-based DFS root?" in the DFS FAQ on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=39465>).

- If the DFS Management snap-in connects to a namespace server that is not running Windows Server 2003 R2 or Windows Server 2003 with SP1, none of the new configuration settings (such as client failback and target priority) can be enabled. Renaming or moving namespace folders will not work, and delegation will not be effective.

Client compatibility for DFS Namespaces

Clients that access namespaces must run one of the following operating systems:

- R2 versions of Windows Server 2003
- Windows Server 2003
- Windows Storage Server 2003
- Windows XP
- Windows Preinstallation Environment (Windows PE) (Can access stand-alone namespaces, but cannot access domain-based namespaces.)
- Windows 2000 Server family
- Windows 2000 Professional
- Windows NT® Server 4.0 with Service Pack 6a
- Windows NT Workstation 4.0 with Service Pack 6a

Additionally, clients that will be configured for client failback must run the following operating systems and hotfixes:

- Windows XP with Service Pack 2 and the Windows XP Client Failback hotfix.
- Windows Server 2003 with SP1 and the Windows Server 2003 Client Failback hotfix.

For more information about the Client Failback hotfix, see article 898900 in the Microsoft Knowledge Base on the Microsoft Web site (<http://go.microsoft.com/fwlink/?LinkId=53202>).

DFS Namespaces size recommendations

- For stand-alone namespaces, we recommend a maximum of 50,000 folders with targets. This recommendation is based on startup times for the Distributed

File System service for large namespaces. For deployments that will approach 50,000 folders with targets, we recommend that you test startup times on your own hardware to determine if startup time is acceptable for service level agreements (SLAs) in your organization.

- For domain-based namespaces, we recommend keeping the namespace object in Active Directory® to 5 megabytes (MB) or less (equivalent to about 5,000 folders with targets). This recommendation is aimed at reducing the time and bandwidth needed to replicate the namespace object using Active Directory replication each time the namespace changes.

DFS Namespaces security requirements and delegation

The following table describes the groups that can perform basic namespace tasks by default, and the method for delegating the ability to perform these tasks.

Task	Groups That Can Perform This Task By Default	Delegation Method
Create a domain-based namespace	Domain Admins group in the domain where the namespace is configured	Right-click the Namespace node in the console tree, and then click Delegate Management Permissions . You must also add the user to the Local Administrators group on the namespace server.
Add a namespace server to a domain-based namespace	Domain Admins group in the domain where the namespace is configured	Right-click the domain-based namespace in the console tree, and then click Delegate Management Permissions . You must also add the user to the Local Administrators group on the namespace server to be added.
Manage a domain-based namespace	Local Administrators group on each namespace server	Right-click the domain-based namespace in the console tree, and then click Delegate Management Permissions .
Create a stand-alone namespace	Local Administrators group on the namespace server	Add the user to the local Administrators group on the namespace server.
Manage a stand-alone namespace*	Local Administrators group on the namespace server	Right-click the stand-alone namespace in the console tree, and then click Delegate Management Permissions .

Task	Groups That Can Perform This Task By Default	Delegation Method
Create a replication group or enable DFS Replication on a folder	Domain Admins group in the domain where the namespace is configured	Right-click the Replication node in the console tree, and then click Delegate Management Permissions .

What Is FRS?

File Replication service (FRS) is a technology that replicates files and folders stored in the SYSVOL shared folder on domain controllers and Distributed File System (DFS) shared folders. When FRS detects that a change has been made to a file or folder within a replicated shared folder, FRS replicates the updated file or folder to other servers. Because FRS is a multimaster replication service, any server that participates in replication can generate changes. In addition, FRS can resolve file and folder conflicts to make data consistent among servers. By keeping files and folders synchronized across servers, FRS enables organizations to increase the availability of data. If one server becomes unavailable, the files are still available, because they exist on another server. Using multiple servers to host data also helps organizations that have offices in multiple geographic locations, because clients can access servers in or closest to their current site and do not need to use expensive WAN links to access data. There are numerous methods for keeping files synchronized on servers. Although SYSVOL requires FRS, DFS shared folders can be kept synchronized by using methods other than FRS, such as manual copying, Robocopy, or other replication tools. FRS provides numerous benefits that other replication methods do not. These benefits include the following:

Authenticated RPC with encryption

To provide secure communications, FRS uses Kerberos authentication protocol for authenticated remote procedure call (RPC) to encrypt the data sent between members of a replica set.

Compression

To save network bandwidth, FRS compresses files in the staging folder by using NTFS compression. Files sent between servers participating in replication, known as replica members, remain compressed when transmitted over the network.

Conflict resolution

FRS resolves file and folder name conflicts to make data consistent among the replica members. If identically named files are created or modified on two or more replica members, FRS uses a “last writer wins” rule; this means that the most recently created or modified version of a file becomes the version that is replicated to the other replica members. If identically named folders are created on two or more replica members, FRS identifies the conflict during replication and renames the folder that was most recently created. Both folders are replicated to all servers in the replica set, and administrators can merge the contents of two folders or take some other measure to reestablish the single folder.

Continuous replication

FRS provides continuous replication, subject to replication schedule, server load, and network load. When a file or folder is changed and closed, FRS begins replicating the changed file or folder to other replica members within three seconds.

Fault-tolerant replication path

FRS does not rely on broadcast technology, and it can provide fault-tolerant distribution by way of multiple connection paths between members. If a replica member is unavailable, the data will take a different route. FRS prevents an identical file from being sent more than once to any replica member.

Replication scheduling

You can schedule replication to occur at specified times and durations as needed by your organization. For example, scheduling replication to occur during evening hours can reduce the cost of transmitting data over expensive WAN links. Replicating data during off-hours also frees up network bandwidth for other uses.

Replication integrity

Files are replicated only after they have been changed and closed. FRS relies on the update sequence number (USN) journal to log records of files that have changed on a replica member. As a result, FRS does not lose track of a changed file even if a replica member shuts down abruptly. After the replica member comes back online, FRS replicates new or updated files that originated from other replica members, as well as replicating locally created or updated files that occurred before the shutdown. This replication takes place according to the replication schedule.

Common FRS Scenarios

FRS is commonly used in the following scenarios:

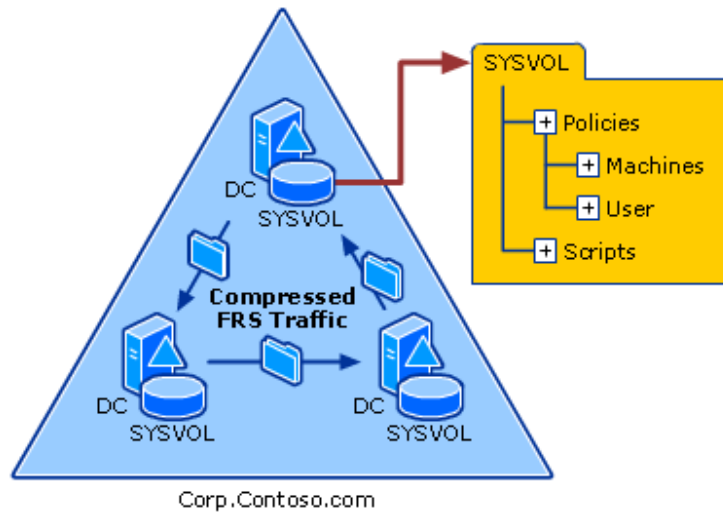
SYSVOL Replication

Every domain controller has a shared folder in its local file system that is the file system component of Active Directory. This shared folder, named SYSVOL, contains files and folders that must be available and synchronized between domain controllers in a domain, including:

- The NETLOGON shared folder, which includes system policies and user-based logon and logoff scripts for non-Windows Server 2003 and non-Windows 2000 network clients, such as clients running Windows 95, Windows 98, and Windows NT 4.0.
- Windows Server 2003 and Windows 2000 system policies.
- Group Policy settings (templates), including Group Policy settings for domain controllers running Windows Server 2003 or Windows 2000.

When you add, remove, or modify the contents in the SYSVOL shared folder, FRS replicates the changed contents to the SYSVOL shared folders on all other domain controllers in the domain. The following figure illustrates how SYSVOL shared folders are kept in sync on domain controllers in the corp.contoso.com domain.

SYSVOL Replication

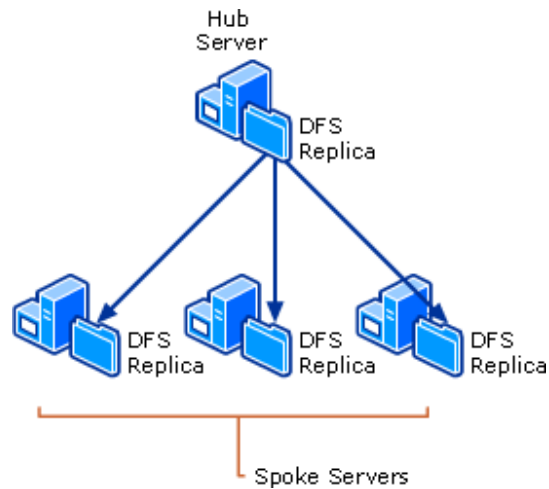


Publishing Applications

FRS and DFS can be combined to distribute and publish applications. DFS allows administrators to group shared folders located on different servers by transparently connecting them to one or more hierarchical namespaces that behave like a single high-capacity hard disk. Users can navigate the namespace without having to know the physical server names or shared folders hosting the data. Administrators can also host data on multiple servers in multiple sites; users see a single copy of the data and DFS can refer users to the closest server to access the data.

When using DFS and FRS to publish applications, an administrator defines the topology used to distribute the data. The topology defines the path along which files are replicated; common topologies for distributing applications are hub-and-spoke, a multilevel tree, and redundant hub-and-spoke. New applications are typically deployed or updated on a single computer (usually the hub or root of the topology or another centrally located computer), and FRS then replicates the files to the spoke computers. The following figure illustrates a hub-and-spoke topology where applications are deployed on the hub and then replicated to the spokes.

Publishing Applications Using a Hub-and-Spoke FRS Topology



Publishing Data

DFS and FRS can also be used to publish data across an organization. Common examples of data include documents, diagrams, and operational procedures.

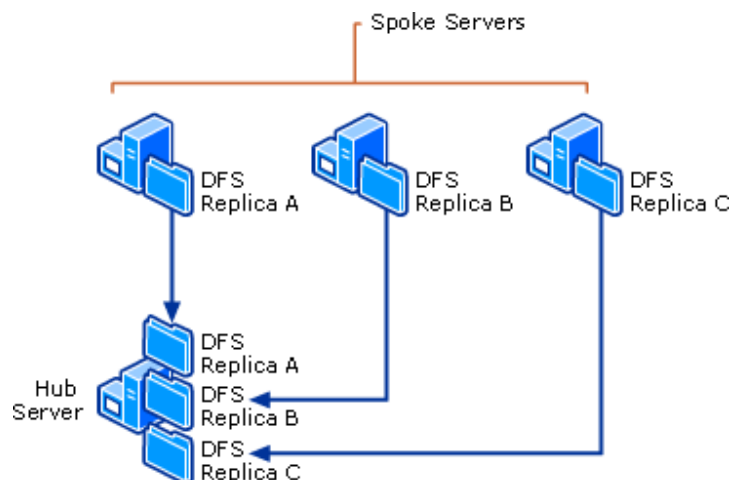
This scenario often uses similar topologies (hub-and-spoke, a multilevel tree, or a redundant hub-and-spoke) as those used for publishing applications.

Another method of publishing data is known as reverse publication. In reverse publication, data flows from a number of different servers to a central server.

Reverse publication is often used to gather log files and reports from individual computers and collect them in one central location. Reverse publication is also used in backup scenarios where data is collected from multiple servers and then backed up from the central server.

The following figure illustrates the reverse publication method.

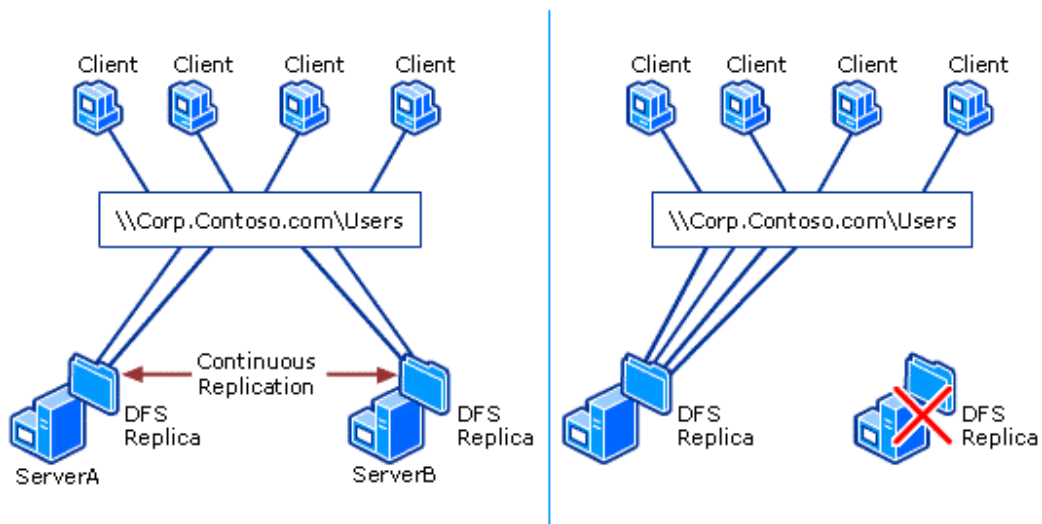
Reverse Publication



Ensuring That Personal Folders Are Available

Organizations that store users' personal folders on file servers can increase the availability of those folders by using DFS and FRS to store the folders on multiple servers. The following figure illustrates a DFS namespace, \\Corp.Contoso.Com\Users, hosted by two servers, ServerA and ServerB. DFS equally distributes the client load between the two servers, and FRS keeps the data synchronized on the two servers. If one of the servers fails, DFS refers clients to the remaining server. Even when one of the servers is unavailable, users can continue to access the \\Corp.Contoso.com\Users namespace.

How DFS and FRS Keep Data Available



Technologies Related to FRS

FRS is closely related to the following two technologies.

Active Directory replication

Active Directory replication is the process by which the changes that are made to Active Directory objects on one domain controller are automatically synchronized with other domain controllers. This replication method does not use FRS and does not include SYSVOL replication. However, FRS replicates SYSVOL by using the connection object topology and schedule that the Knowledge Consistency Checker (KCC) creates for Active Directory replication. In addition, FRS has its own Active Directory objects that are replicated using Active Directory replication.

DFS

FRS can be used to keep data in DFS shared folders synchronized among replica members. However, DFS and FRS are two separate technologies, and DFS does not require FRS. You can use other replication methods, such as manual copying, the Windows Resource Kit tool Robocopy, or other replication tools to keep DFS shared folders synchronized. Conversely, if you want to use FRS to keep data in shared folders synchronized, you must use DFS.

FRS Dependencies

FRS has the following dependencies:

- **Active Directory replication.** FRS requires that Active Directory replication is working properly so that FRS Active Directory objects reside on all domain controllers in the domain.
- **DFS.** If you want to use FRS to keep data in folders synchronized on multiple servers, you must first set up a DFS namespace. (This dependency is not applicable to SYSVOL.)
- **DNS.** FRS requires that DNS is properly designed and deployed so that FRS can correctly resolve DNS names of replica members. (DNS is also required for Active Directory replication.)
- **Kerberos authentication.** FRS requires that Kerberos authentication is properly designed and deployed.
- **NTFS.** To detect changes to files and folders, FRS relies on the USN journal on NTFS volumes. FRS is not supported on FAT file system volumes.
- **Remote procedure call (RPC).** FRS requires Internet Protocol (IP) connectivity and the Remote Procedure Call (RPC) for communication between replication partners and communication with domain controllers.