



# Microsoft DFS: Leveraging the Benefits and Filling the Gaps

*Sponsored By:*





# Microsoft DFS: Leveraging the Benefits and Filling the Gaps

## Table of Contents:

[Introduction](#)

[Why Use Microsoft DFS?](#)

[Problems with FRS—The Advantage of DFS R2](#)

[DFS R2 Improvements—Namespaces and Replication](#)

[The DFS R2 Limitation: No File Locking](#)

[Filling the File Locking Gap](#)

[Resources from Peer Software, Inc.](#)

## Introduction

Networking has made it possible for organizations to locate offices and employees anywhere. Working with partners, selling products, and providing service can also be accomplished for any connected location. As long as files can move from point A to point B, there's nothing standing in the way of productivity.

Of course, organizations with distributed workforces, sales regions, and partner locations have many files moving from point A to point B. Those files could contain any kind of data: copy, financials, software code, graphics. And many times, files are works in progress, travelling the network from one member of a committee to another, from an editor to a writer and back again, or from a supervisor asking for spreadsheet updates to the accountant who will make them.

IT management needs to ensure the organization that data and files will move securely from point to point. It must also find efficient ways to serve the data and enable people to share files. Without such efficient systems, productivity is compromised by file check-ins and check-outs and the need to know where files currently reside.

Microsoft Distributed File System (DFS) enables IT managers and end-users in medium and large enterprises access and manage files that are distributed across the network. DFS replication services allow files to be shared without emailing from person to person. There is no need to search for the latest version, the last recipient, or worry about the network's capacity to move a large file.

However, Microsoft DFS does present some limitations, most notably the inability to lock files. Without file locking, users that access a file simultaneously might inadvertently overwrite one another's changes or modify contents that are still being formulated. The time needed to identify and resolve data conflicts can negate the benefits of file sharing using Microsoft DFS.

PeerLock from Peer Software adds file locking to Microsoft DFS, filling a critical gap and allowing organizations to use file sharing to the greatest benefit. Used in combination with PeerSync, PeerLock also enables organizations to realize true distributed collaboration.

## Why Use Microsoft DFS?

Distributed System File Overview (Microsoft TechNet 2005) offers this description of DFS:

With Distributed File System (DFS), system administrators can make it easy for users to access and manage files that are physically distributed across a network. With DFS, you can make files distributed across multiple servers appear to users as if they reside in one place on the network. Users no longer need to know and specify the actual physical location of files in order to access them.

For example, if you have marketing material scattered across multiple servers in a domain, you can use DFS to make it appear as though all of the material resides on a single server. This eliminates the need for users to go to multiple locations on the network to find the information they need.

---

---

The article goes on to list these reasons for using DFS:

- You expect to add file servers or modify file locations
- Users who access targets are distributed across a site or sites
- Most users require access to multiple targets
- Server load balancing could be improved by redistributing targets
- Users require uninterrupted access to targets
- Your organization has web sites for either internal or external use

That list applies to almost any organization in which people in different locations collaborate. Obviously, Microsoft DFS has broad application across vertical markets and organization sizes.

## **Problems with FRS—The Advantage of DFS R2**

Managing distributed files requires replicating them for use by multiple users in multiple locations. Microsoft DFS uses a technology called File Replication Services (FRS) to replicate files between servers. While FRS certainly does accomplish replication, it presents some real problems in terms of efficiency and, because Microsoft has moved to a completely new codebase, is difficult to maintain.

*In "Leveraging DFS in Windows Server 2003 R2" Gary Olsen points that Microsoft Windows Server 2003, offers some real advantages over its predecessor, Windows 2000 File Replication Service (FRS) including remote differential compression and a very easy migration path.*

---

Since Windows Server 2003 has been out for a while now, it has lost a bit of its luster in the eyes of the media, which has turned most of its attention toward the upcoming release of Longhorn.

What gets lost, however, is that Windows Server R2 is really a sneak peak into Longhorn, as all the new functions introduced in R2 will be available in the new operating system, including new Distributed File System components and DFSr.

### **The birth of DFSr**

Perhaps the most significant of these features are the new Distributed File System components, including the new replication engine DFSr.

There is little doubt that File Replication Service (FRS) was poorly implemented in Windows 2000. Administrators



## ***Increase Productivity Eliminate Version Conflicts when using DFSr***



### **PeerLock Adds File Locking Capabilities to your MS DFS Replication environment to eliminate version conflicts for multi-location teams**

PeerLock Server offers network file locking while local files are in use and integrates seamlessly with Microsoft DFS Replication.

This innovative program ensures that when a user is modifying a file, no other user will be allowed to make changes to that file on any machine that the user has chosen to lock

#### **PeerLock Server gives you:**

- ◆ Real-time detection of file use
- ◆ Immediate remote file locking
- ◆ MultiThreading for robust performance
- ◆ Easy installation and unobtrusive operation
- ◆ Failed connection retries
- ◆ Cross domain file locking
- ◆ Log file reporting

#### **Free 15 day evaluation:**

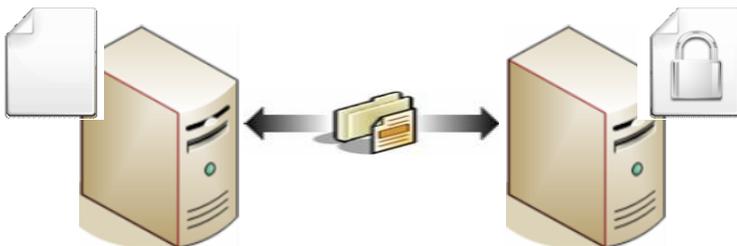
To download a free 15 day evaluation, please visit:

[www.peersoftware.com/trial\\_download.asp](http://www.peersoftware.com/trial_download.asp)

#### **For more information, contact us:**

Phone: 631.979.1770  
Email: [sales@peersoftware.com](mailto:sales@peersoftware.com)  
Address: 4155 Veterans Highway  
Suite 12  
Ronkonkoma, NY 11779

URL: [www.peersoftware.com](http://www.peersoftware.com)



literally lived from hotfix to hotfix trying to patch FRS to make it work. Windows Server 2003 made some significant changes to FRS in an attempt to make it more tolerant of error conditions, but there was still no real fix for anything. It's possible that Microsoft simply gave up trying to patch FRS, because it rewrote the replication engine from scratch with R2. Thus, through the use of better design criteria and years of FRS experience to temper the decisions, DFSr was born.

DFSr has several key features, but the most important one is, arguably, *remote differential compression (RDC)*. RDC replicates changes to a file, rather than the entire file itself as FRS did. For example, if I had a 3 MB Word document, and you edited a couple of lines, FRS would replicate the entire 3 MB file. RDC, on the other hand, replicates only the changes—in this case, just a few bytes. Therefore, instead of taking several minutes for FRS to replicate this change, DFSr will replicate it in a few seconds (depending on the network speed).

Unfortunately, Microsoft only had time to add DFSr to replicate DFS data, and not the System Volume (SYSVOL), so SYSVOL still uses FRS for replication.

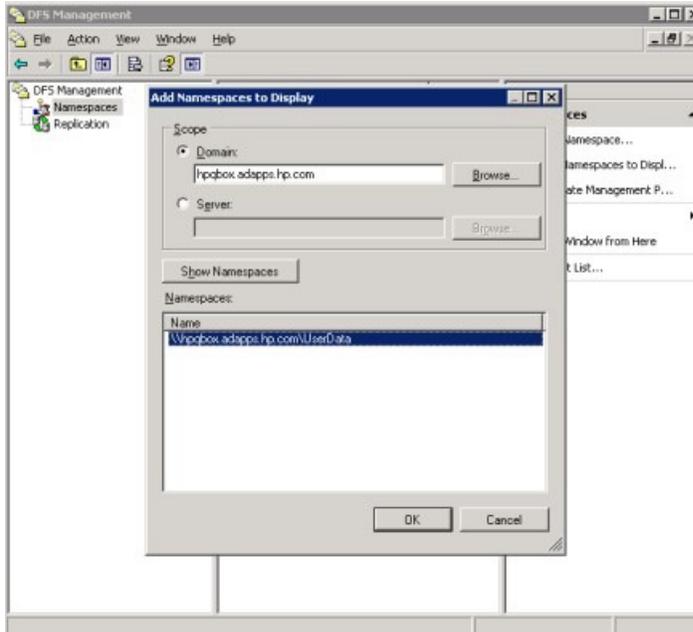
## Migrating to R2/DFS

Migrating an existing Windows 2000 or 2003 Distributed File System structure to the DFS configuration couldn't be easier. Here is what you need to know:

1. Upgrade DFS servers to R2. You don't need to upgrade all servers to R2—just the ones hosting the DFS namespace you want to migrate to R2.
2. Note that installation of R2 requires a schema upgrade, so plan for that.
3. Even after the R2 upgrade, your DFS namespace has not changed to the new R2 DFS namespace. You can still use the old DFS admin tool. Nothing has changed for DFS.
4. After the R2 upgrade, you need to go into Add/Remove Programs, open Windows Components and install DFS, just like you would any other component, such as DNS for instance. Even after installing DFS, you'll still be using the old DFS namespace and FRS for replication.
5. These steps must be performed on all servers that host or will host the DFS namespace.
6. Once the R2 upgrade and DFS installation have been completed, it's time to migrate the old DFS namespace into the new DFS and configure DFSr replication.

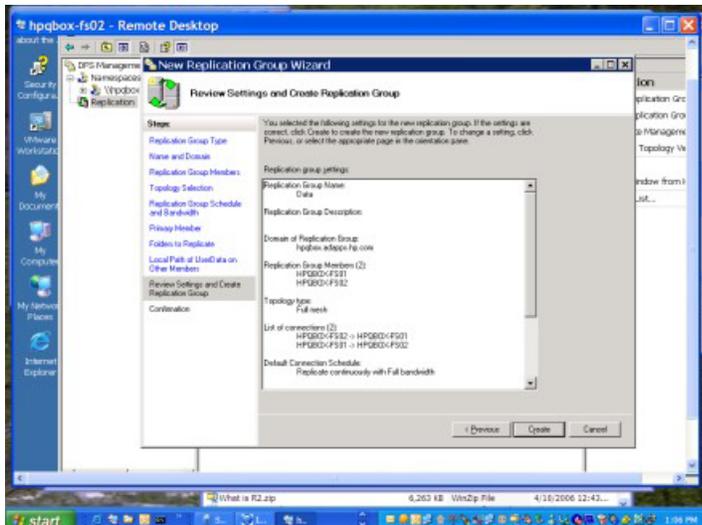
To do this, open the new DFS management tool and you'll see the *Namespaces* and *Replication* icons in the left pane. Right-click on *Namespaces* and select *Namespaces to Display*. Just like the old DFS snap-in, it will then display all the defined namespaces, as shown in Figure 1. This is similar to what you would do prior to R2 to add an existing DFS namespace in the domain to the DFS snap-in. It will show all existing namespaces—just select one to load it into the new snap-in.

**Figure 1**



7. Right click on *Replication* in the left-hand pane and select *Replication Wizard*. This wizard is extremely easy to use. There is no need to read white papers to figure out what a link, target, link target or root target is. I was able to configure replication without reading any help files, calling Microsoft Support or asking anyone else for help. This is especially true if you have previous DFS experience. Choose the replication configuration you want and complete the wizard. Figure 2 is a summary of the replication selected.

**Figure 2**

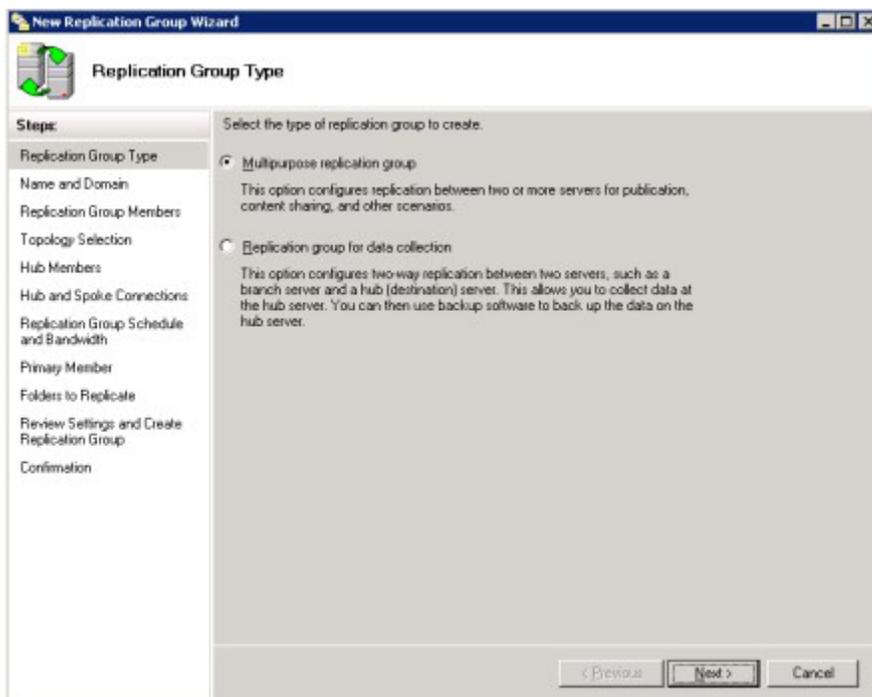


8. That’s it! You have just moved your DFS structure to the new R2 environment, including using the DFSr replication.

Now that you are using the new Distributed File System, you will be able to note several new features right away, including:

- *Better replication performance using DFSr.*
- *New options in configuring replication.* As shown in Figure 3, there is the Data Collection option, which is used for replicating data from branch sites to a hub site. The other is the “Multi-purpose Replication group,” which is for normal DFS sharing.

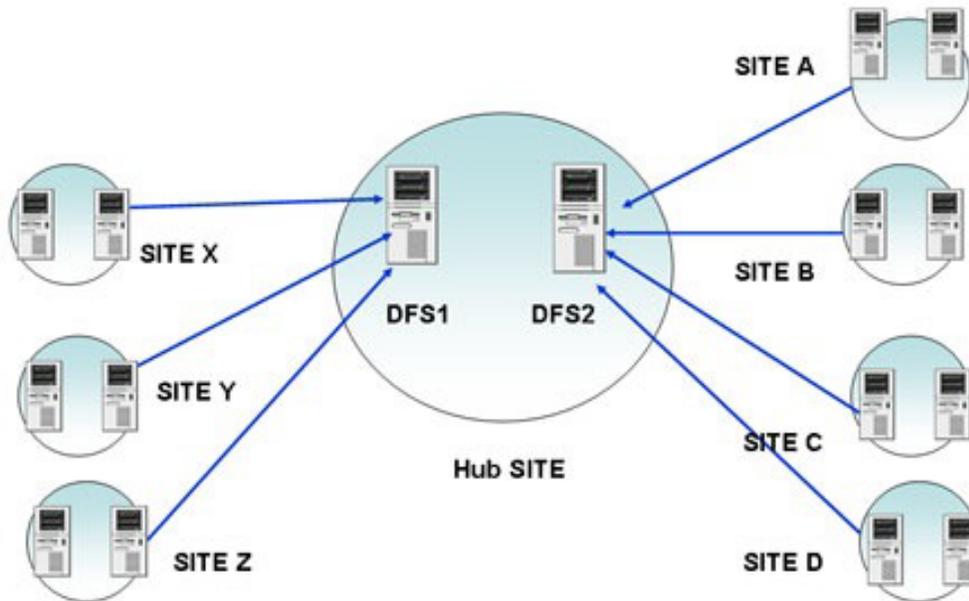
**Figure 3**



*Easy to understand terminology.* Just click on the name space and look in the upper-center pane.

Example: I saw one situation where DFS had been constructed masterfully to back up data from the 100 branch sites to one of two servers in a hub site. To do that, the admins built a server for DFS in each of the branch offices. Using Windows 2003 DFS, they created links for each branch office with only one of two hub servers, as shown in Figure 4.

Figure 4



Each link had only two target servers—one in the branch site and one in the hub site. This was done to take advantage of Microsoft’s philosophy of branch office design. That is, replicate the data to the hub site and back up the hub.

However, the administrator reported a performance problem on one site that had a 62 GB share size, which is close to Microsoft’s recommended maximum of 65 GB. When I recommended that they move to R2, they objected, saying they couldn’t afford to upgrade all of the machines. My recommendation was to add an additional server at the remote site, add a DFS share to the new server and split the load between the servers. The next step would be to upgrade those two resources (site DFS servers in the remote site) to R2, and upgrade the hub server they connect to via DFS.

In this example, we would have also had to create a new namespace with the R2 servers. We could remove that link from the old DFS snap-in and create a new namespace in R2, then configure replication using R2’s Data Collection mode. This would provide the benefits of R2 to a problematic site without having to do the upgrade on all servers.

The DFS functionality in the Windows Server 2003 R2 release has some powerful benefits that improve performance of DFS replication—benefits that will also be included when Microsoft releases Longhorn. The really good news is that Longhorn will improve on this by using the DFSr replication engine to replicate SYSVOL as well.

---

## DFS R2 Improvements—Namespaces and Replication

Laura Hunter also lists a variety of R2 enhancements in her February 2007 article “Not Your Father’s DFS.” Key to the DFS R2 functionality set are namespaces and replication—two feature areas that offer real distributed file management power and represent significant advances over previous versions.

In R2, Microsoft split the DFS service into two components: DFS-Namespaces (DFS-N) and DFS-Replication (DFS-R). This division allows you to make a more granular decision about the services you deploy on your file servers. If you only use a unified namespace, you can skip installing the DFS-R component unless (and until) your environment expands to the point that it requires replication capabilities.

The DFS Namespaces feature in R2 offers the following updated capabilities:

1. **Target priority.** If DFS detects that a particular link target or folder target is inaccessible, it will automatically route clients to another target server. In previous versions of the server OS, if you had multiple link targets specified for a particular link, you could not specify the order in which referrals should take place. In R2, you can specify a priority list of targets that the client will be referred to.
2. **Client failback.** In previous versions of DFS, if a client was routed past an unavailable link target to another in the list of link targets, the client would continue to use that server until the client was rebooted or until its DFS referral cache was cleared. In R2, clients can fail back to a preferred local server once its availability has been restored. However, client failback is available only for clients running Windows XP SP2 or Windows Server 2003 SP1 (including R2). Both OSes require a hotfix that’s available from Microsoft.
3. **Delegation of authority.** With DFS-Namespaces in R2, you can delegate the ability to create namespaces, as well as the ability to administer existing namespaces, by setting the necessary permissions within Active Directory (for a domain-based namespace) or in the server Registry (for a standalone namespace). By default, you need Domain Admin rights to manage domain-based namespaces or to be a member of the local administrators group of the server that’s hosting a standalone namespace.

### DFS-R: This *really* isn’t your father’s DFS

In R2, it’s in the realm of DFSReplication (DFS-R) that the new DFS really begins to shine. A new replication algorithm provides incredible performance gains for bandwidth challenged environments such as a branch office separated from corporate headquarters by a low-speed or heavily used WAN link.

Prior to R2, DFS used the File Replication Service (FRS) to replicate files between multiple link targets. FRS is the service used to replicate the information stored in the AD SYSVOL share: logon/logoff scripts and Group Policy Objects.

FRS uses RPC over TCP/IP to replicate files within a single site as well as between sites; FRS creates its own replication topology with its own schedule and connection objects that are controlled separately from AD replication.

FRS will trigger replication whenever a file is closed, with changes held in a 3-second aging cache to allow for files that are changed frequently. Once this 3-second “waiting period” is up, the FRS service on the server hosting the changed file notifies its FRS replication partners, and the file is replicated across the FRS replication topology. For small files stored on lightly used servers, this process works quite well. But because FRS traffic is not compressed even when traversing site boundaries, replication of large files can be a tricky process, one that often creates replication errors or inconsistencies.

DFS in R2 changes all of this by introducing a replication algorithm called Remote Differential Compression (RDC). RDC breaks files up into small chunks, then replicates only the individual chunks of a file that have changed from one replication cycle to the next.

Consider a Microsoft Word file that contains the line of text: The quick brown fox jumps over the lazy white dog. If someone changed the sentence to read “The slow black fox,” then RDC would replicate only that particular chunk, rather than sending the entire .doc file across the wire. RDC does this by computing MD4 hashes of these small chunks of files, then comparing those hashes between servers that are attempting to replicate.

If a particular chunk of a file has changed, the MD4 hash of that chunk will change while the hashes for the remaining chunks in the file remain the same. This allows an RDC-enabled server’s replication partner to request only those chunks whose hashes have changed since the last time replication took place. For larger files that only need to replicate small changes, this will reduce replication time drastically, while improving performance for your users.

Imagine a 4 MB Visio document in which you need to change the title of one or two sections. In the FRS world, that would prompt the entire 4 MB file to replicate. However, RDC needs only a few seconds to replicate the changed sections of the file. For environments with branch offices to support, particularly where bandwidth is at a premium, DFS-R in R2 can more than justify making the move to the new OS.

Because DFS Replication is triggered on file close, it’s not efficient for replicating files that are always locked and in use, as in the case of a database or another file used by an “always-on” service. Nor can you use DFS-R to replicate the AD SYSVOL share; Logon scripts and Group Policy Objects still need to be replicated via FRS. However, FRS and DFS-R can co-exist comfortably on the same server.

Now let’s examine some of the other improvements in DFS Replication in Windows Server 2003 R2:

1. Bandwidth throttling and replication scheduling. To gain more control over the use of your bandwidth, you can specify replication schedules similar to those you’d set up between sites in AD. You can specify these schedules for an entire replication group or create a custom schedule for an individual replication connection. You can also cap on the amount of bandwidth that DFS-R replication can take up.
2. Support for replication groups. You can configure one or more sets of data and servers as a replication group with a common configuration for replicated folders, replication schedules and bandwidth throttling. Each DFS server can support a maximum of 256 replication groups, and each of these groups can contain up to 256 replicated folders.



## Ensuring Comprehensive Enterprise Data Availability and Data Collaboration Solutions

File Synchronization  
File Replication  
File Backup  
File Collaboration

Ensuring Comprehensive Enterprise Data  
Availability since 1993 ➔



Peer Software provides cost effective, high-performing solutions that ensure data availability and data collaboration through disk to disk synchronization and replication of all data within an organization.

Through flexible licensing terms for servers, workstations, and laptops, we can provide a solution to meet the needs of any customer. Over 8,000 customers world-wide have used Peer Software to meet the growing data demands of today's business.

### Find a solution to meet your needs:

- ◆ DFS File Locking
- ◆ Multi-Office File Collaboration
- ◆ Full Mesh Server Mirroring
- ◆ Centralized Server Backup
- ◆ Centralized Laptop and Desktop Backup
- ◆ File Distribution
- ◆ Database Backup

### Free 15 day evaluation:

To download a free 15 day evaluation please visit:

[www.peersoftware.com/trial\\_download.asp](http://www.peersoftware.com/trial_download.asp)

### Key features that you can rely on:

- ◆ Real-Time or Schedule Event Processing
- ◆ Centralized Configuration and Monitoring
- ◆ Byte-level Replication
- ◆ Multi-Threaded Performance
- ◆ Bandwidth Throttling
- ◆ UNC Path and TCP/IP Protocol Support
- ◆ Database / open file backup support
- ◆ Advanced Reporting

### For more information, contact us:

Phone: 631.979.1770  
Email: [sales@peersoftware.com](mailto:sales@peersoftware.com)  
Address: 4155 Veterans Highway  
Suite 12  
Ronkonkoma, NY 11779

URL: [www.peersoftware.com](http://www.peersoftware.com)

3. Collecting data for backup purposes. You can use replication groups to collect data from branch sites to perform centralized backups. Rather than relying on remote sites to maintain their own backup hardware and perform their own backups, you can create a separate replication group to replicate their data to a central location. By disabling replication from the hub site back to the branch server, you'll create a "one-way" replication agreement that prevents any inadvertent changes made at the backup site from replicating back to the remote server. DFS-R can replicate data across multiple forests within the same forest; you're not restricted to replicating within a single domain.
4. Cross-file rdc. This takes the performance improvement of RDC to the next logical level. Say you have a file stored in a DFS namespace called 2006 Board of Directors.doc detailing the names and biographical information of your company's board for that year. You need to create a similar file for the 2007 board, so you save the 2006 file as 2007 Board of Directors.doc and make a few changes to reflect two new board members.

Now there's a new file that needs to be replicated within the DFS namespace. But is it really brand new? By using cross-file RDC, DFS can use the contents of the 2006 Board of Directors file to seed replication for the new file, using the "chunking and hashing" method already described to send over the wire only the information that's different between the two files. (This feature is possible because comparing the MD4 hashes created by two files is far more efficient than comparing the actual contents of the files.)

5. File and subfolder filters. You can specify individual subfolders or filenames that should not be included in DFS Replication, either by explicitly listing the name of the file or folder or by using the \* wildcard symbol. By default, DFS-R will not replicate any folder that begins with the tilde (~) character, as well as any files with a .TMP file extension.

Other files and file types that will always be excluded from DFS Replication include:

- any EFS-encrypted files;
- any file that has had the temporary attribute set;
- any reparse points used by Single Instance Storage or Hierarchical Storage Management (The reparse points used by DFS itself are not affected by this.); and
- any NTFS-mounted drive paths where you've added a new drive to a system and assigned its space as a folder within an existing drive letter, rather than assigning it a drive letter of its own.

## The DFS R2 Limitation: No File Locking

Despite the great advantages Microsoft DFS brings to distributed organizations that rely on file sharing and collaboration, there are still major gaps in the technology that require attention.

Organizations typically begin to use DFS R2 for branch office backup. They quickly move to branch office mirroring. At that point, administrators discover that DFS R2 lacks file locking capabilities. Without file locking, files routinely fall out of sync as changes are made simultaneously or in between replication intervals.

---

Laura Hunter details the file locking problem in "Not Your Father's DFS."

Perhaps the most important limitation is this: DFS Replication is suitable for environments that can tolerate a certain loose consistency between different copies of a particular document. Even given the performance enhancements of the Remote Differential Compression replication, there will still be a slight amount of inconsistency between servers in a replication group as a change is replicated between them.

Consider this situation: You've configured a replication schedule that only allows for DFS Replication between 11 p.m. and 5 a.m. for a site whose WAN link is fairly saturated during the day. If a user makes a change to a file stored on ServerA at 1 p.m. on a Tuesday, and another user accesses the copy of that file that's stored on ServerB at 2 p.m. on the same day, the second user will not see the changes that were made on ServerA. The changes have not replicated yet.

Even if you allow DFS Replication to take place 24x7, limitations in network transmission speed mean that ServerA's copy of a document might not be precisely in sync with ServerB's copy at any given second. If you're working with documents that have zero tolerance for this type of loose consistency, you might want to consider a document collaboration solution (one that allows for document check-in and check-out), such as Microsoft SharePoint. But in most cases, the replication improvements offered by R2 DFS will provide an easy way of synchronizing files across multiple locations.

But what happens if two people create a replication conflict by managing to modify the same file at the same time while working from two separate servers? Similar to AD replication, the RDC algorithm resolves conflicts by taking the "last writer wins" approach: Whichever file was modified most recently is the one that will win the conflict and be replicated throughout the namespace. The "losing" file will be renamed and stored in a Conflict and Deleted folder on the server that processed the conflict. Details of the file will be stored in a ConflictandDeletedManifest.xml file. This folder has a default quota of 660 MB and will be automatically purged when its size reaches 90% of that limit.

---

The "last writer wins" approach is a fairly unsatisfactory way to determine which file changes will take precedence. Even if two colleagues are not editing a file simultaneously, DFS Replication methodology can lead to conflict. DFS Replication works on a single thread "pull" process; synchronization tasks queue up and can create a backlog that keeps changes made in one location from being immediately replicated in another. The time delay creates an even greater probability of version conflict.

---

## Filling the File Locking Gap

Realizing the full benefit of Microsoft DFS requires filling the file locking gap. The best way to eliminate version conflict when using DFS is to add a true file locking solution. The solution should include real-time detection of file use and immediate remote locking that assures that when a file is open at location A, all other versions (for example, local copies at branch offices) are locked down. Colleagues in the branch offices will see that the file is in use and understand why they are temporarily locked out. When the file is closed by the editor in location A, it is unlocked, released, and ready for synchronization.

### PeerLock from Peer Software, Inc.



Peer Software Inc. has developed PeerLock Server for use as a stand-alone file locking solution for use with Microsoft DFS. PeerLock’s core functionality ensures that when a user is modifying a file, no other user will be allowed to make changes to that file on any machine that the user has chosen to lock. PeerLock includes real-time detection of file use and immediate remote locking. Operation is unobtrusive; your users will focus on their files, not the file locking system. In addition, PeerLock supports any number of geographically disperse office branches and servers. File locking and release is multi-threaded (rather than single-threaded which is the case with DFS) to prevent queuing and eliminate associated file conflict.

PeerLock also eliminates several critical file management barriers. Files can be locked in any location even if the location is on a different domain. Seamless collaboration is facilitated by reducing file release intervals—longer intervals increase the risk of version conflict. Locking options offer more precise control and increase visibility into locked files across the collaborative environment.

## Microsoft DFS with PeerLock

Microsoft DFS is a key technology for companies that need to facilitate collaboration between distributed employees, remote offices, and geographically disperse partners. But, it cannot protect the benefits of file sharing when more than one person is using the file.

PeerLock from Peer Software allows Microsoft DFS users to detect when files are in use and lock them—no one can overwrite someone else’s work or introduce conflicts. PeerLock makes file sharing seamless and productive, protecting your overall Microsoft DFS investment.

## Resources from Peer Software, Inc.



[Case Study for PeerSync Collaboration Package for inter-office file collaboration.](#)

[Web Demo for configuration and installation of our PeerSync Collaboration Package.](#)

[Learn about how PeerSync Workstation enables centralized enterprise scale laptop backup for mobile workers.](#)

### About Peer Software, Inc.

Peer Software develops powerful, cost-effective, and easy-to-use file and network management software allowing organizations to seamlessly, efficiently, and reliably protect their digital assets. The company's flagship products are PeerSync and PeerLock. PeerSync software offers enterprise wide data availability (backup, synchronization, replication, and data distribution), and is designed for servers and workstations. PeerLock software offers file locking capability for geographically dispersed project teams and multi-location companies to ensure data consistency. Close to half of current Fortune 100 companies entrust critical file management to Peer Software.

[www.peersoftware.com](http://www.peersoftware.com)