# Microsoft DFS Replication vs. Peer Software's PeerSync & PeerLock

## Contents


Why Replication is Important . . . . . 2
The Original Purpose for MS DFSR . . . . . 2
Best Scenarios for DFSR . . . . . 3
When DFSR is Problematic . . . . . 4
The Trouble with Troubleshooting . . . . . 5
Technology Choices . . . . . 7
PeerSync - A Best-of-Breed Replication Technology . . . . . 7
PeerLock Completes the Solution . . . . . 9
Feature Comparison . . . . . 10
Summary . . . . . 11


www.PeerSoftware.com

When a technology is adopted and implemented by an organization it has proven that it meets an original set of requirements identified by key stakeholders. As time progresses, it is human nature to attempt to apply this same technology to new applications not anticipated by the technology developer, only to experience unforeseen consequences. An example of this phenomenon is Microsoft Distributed File System Replication (DFSR).

In this document we will help the reader understand the strengths and limitations of DFSR and demonstrate how Peer Software's PeerSync and PeerLock technology (PeerSync Collaboration Edition) has evolved to become the standard for enterprise file replication and synchronization.

## Why Replication is Important

Given today's explosion of enterprise data, the challenge of enabling data access across the enterprise (and by extension across the world) is becoming more problematic. Even small companies are now often engaged in transcontinental collaborations for mission critical projects. File sizes are on the increase while at the same time teams are becoming more far flung due to the global nature of business, in many cases connected by a costly yet fragile Wide Area Network (WAN) infrastructure.

For all organizations the imperative is to increase efficiency. The easiest and most effective way to do this is to first focus on the most expensive resources of an organization, which typically is the time of skilled employees. If you can recover a Full Time Employee's (FTE) worth of time due to automation and optimization that is sweet music to the bottom line. All the solutions discussed in this document seek to address realizing this return on investment.

The second most expensive IT resource, after the staff, is likely bandwidth between distant locations. This is especially true when dedicated WANs are required. If you doubt this, just think of your last system deployment and consider whether anyone would really bat an eye if you told them that the server needed twice the amount of RAM, or disk capacity, or even processor speed for things to work effectively. In almost every case the answer would be no problem since the incremental cost is low, but if you asked to double the bandwidth for a major dedicated WAN the project would probably have been dead on arrival.

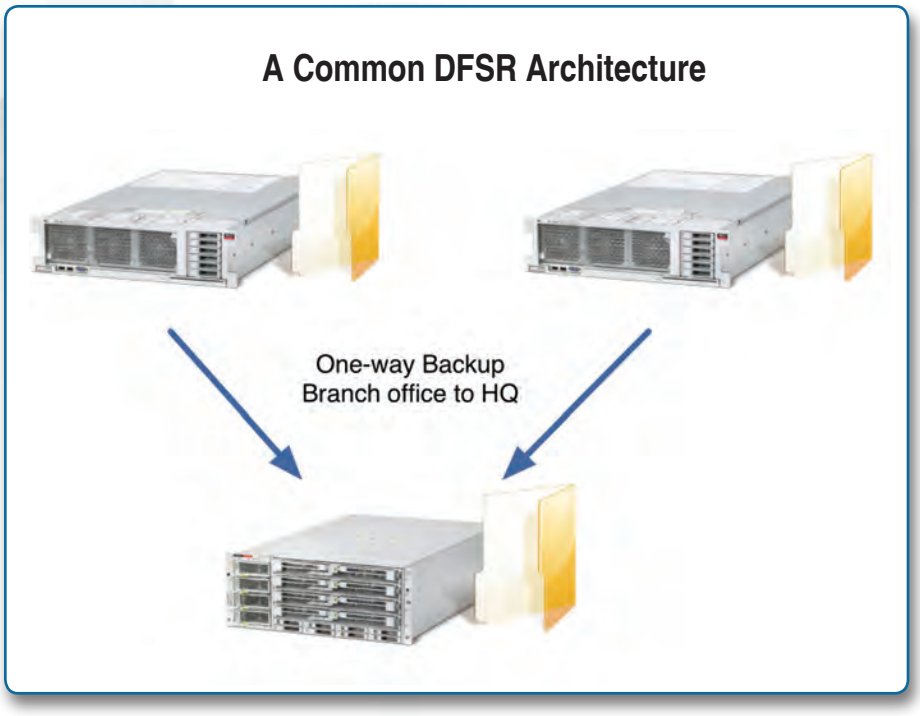
Finally, storage, due to its high cost of management, is next on the list of costly IT resources. Replication done well increases your costs in storage, but this expense will almost always be made up many times over by the savings in manpower and bandwidth.

## The Original Purpose for MS DFSR

Microsoft Distributed File System Replication (DFSR) was originally designed to provide an easy and integrated way of distributing a limited amount of important documents for read-only access to branch offices or vice versa.

Soon, administrators started to use DFSR to not only distribute documents but also to backup data from branch offices to the HQ server. DFSR worked well for one-way backup over wide area networks for small environments where backing up more than ten thousand files became completely unmanageable to handle manually.

By providing an automated and efficient mechanism for moving files over the WAN as needed, DFSR was revered by many administrators who used it for this purpose.




The real world challenge with DFSR is that replication tasks are bidirectional in nature and the administrator has to make sure that data on the target system can only be accessed read-only.

## Best Scenarios for DFSR

What are people doing with DFSR today? As mentioned earlier, DFSR serves quite well in some basic scenarios such as providing backup functionality across WANs. The ability of DFSR to only replicate file changes can minimize traffic over WAN links, but keep in mind that in this scenario file copies are for backup purposes, meaning they are either read-only or just for disaster recovery if the source fails in some way.

Another scenario where DFSR makes sense is providing bidirectional data synchronization where file locking is not required. This last part is significant in that most users do not recognize a need for file locking until they are in production and encounter problems without it.

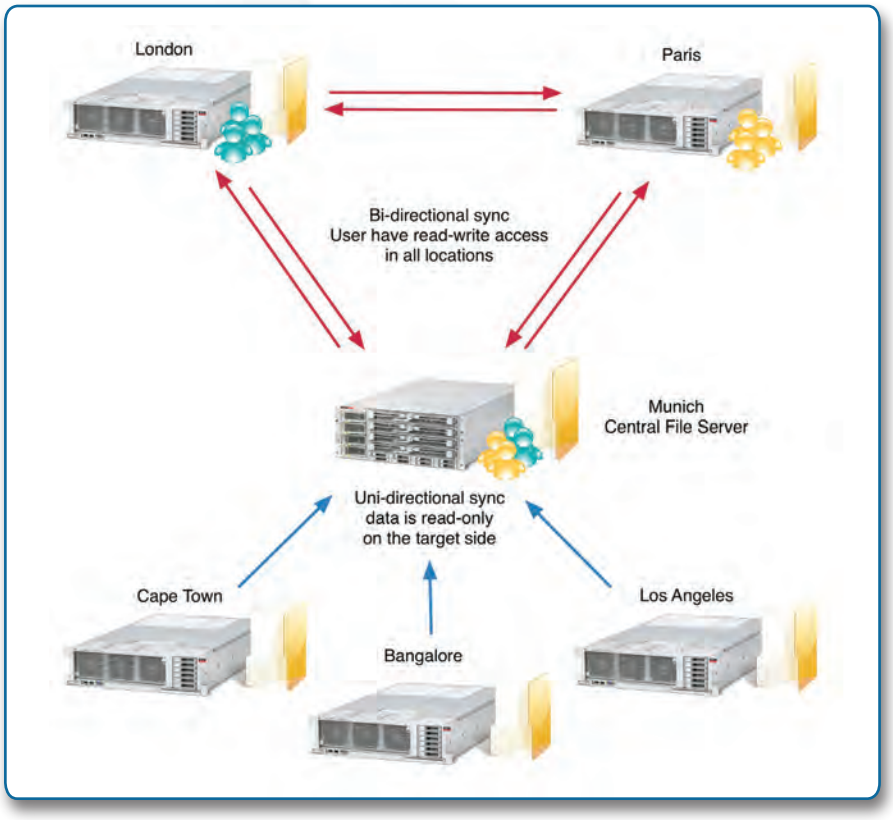
## When should I not use DFS Replication?

"Do not use DFS Replication in an environment where multiple users update or modify the same files simultaneously on different servers. Doing so can cause DFS Replication to move conflicting copies of the files to the hidden DfsrPrivate\ConflictandDeleted folder."

*Source: DFS Replication FAQ, Microsoft TechNet (updated March 30, 2011)*

## When DFSR is Problematic

DFSR is built into the Microsoft server operating system which is a mixed blessing. It is convenient since it is there without needing an install, but it means that you are limited to what is included in the OS version you have. In mixed environments where both Windows 2003 and Windows 2008 are deployed, and hosting shares need to be synchronized, the lowest common denominator is in effect meaning that your implementation is only capable of taking advantage of the Windows 2003 features even if you have Windows 2008 R2 servers also in the mix.




The biggest problems often take time to really show themselves. For example, the lack of file locking allows conflicts when using bidirectional synchronization, but it almost never comes up during proof of concepts or testing. However, in a large environment these conflicts happen so often that the cost outweighs the value of having the system perform the synchronization.

## The Trouble with Troubleshooting

DFSR is not a transparent technology. It performs its functions and provides very little feedback related to how things are working. An example of this can be found in the event logs where reported errors are very common and do not help much or pertain to vague connectivity issues, which often lead you astray when troubleshooting. For example, there are multiple reported incidents that when faced with a large file (gigabytes) to replicate, DFSR tends to lock up and often will not recover even if the large file is deleted. The recovery attempt typically causes error 4412 to start showing up indicating "a file was changed on multiple servers". At this point there is no way to get it back to working condition without starting your setup over from scratch.

In larger environments, such as those with a dozen or more locations/servers replicating data, it can be a common event for a file to be changed in multiple locations depending on what you are replicating. This situation is noted by DFSR in the Windows Application Log.

If there are too many sharing violations, DFSR can start spending more time retrying locked files than it does replicating unlocked ones and this will impact performance. A large number of event log entries with IDs 4302 and 4304 indicate this problem and point to a real need for a scalable enterprise file locking technology to complement the replication system.

There have been improvements in Windows Server 2008 R2 including a dedicated log for DFSR events, the Health parameter for the DFSRAdmin.exe utility, and the DFSRDiag.exe utilities, but lacks a way to lock files to prevent multiple simultaneous edits.

Below is an example of the type of information that DFSRDiag.exe provides. As you can see the output includes the list of the files updated as part of the replication with a great deal of internal data that is not helpful such as IDs and various GUIDs. If you wade through this output you can determine what is going on, but at a high cost in productivity.

DFSR logging and reporting is designed more for developers than time constrained administrators.

## A dfsdiag Example

```
Active inbound connection: 1
  Connection GUID: F874A1D0-77DF-4521-BCDE-0B09E3006409
  Sending member: CONTOSO-HUB
  Number of updates: 36

  Updates being processed:
      [1] Update name: NlsData0049.dll                    (Downloading)
        Path                  :
        UID                   : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1343
        GVSN                  : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1343
        Parent UID            : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v45
        Replicated folder     : 7C764107-1521-4263-B827-04B0155C6067
      [2] Update name: NlsData004b.dll                    (Downloading)
        Path                  :
        UID                   : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1345
        GVSN                  : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1345
        Parent UID            : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v45
        Replicated folder     : 7C764107-1521-4263-B827-04B0155C6067
...
      [15] Update name: NlsData0010.dll                   (Downloading)
        Path                  :
        UID                   : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1323
        GVSN                  : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1323
        Parent UID            : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v45
        Replicated folder     : 7C764107-1521-4263-B827-04B0155C6067
      [16] Update name: NlsData0045.dll                   (Downloading)
        Path                  :
        UID                   : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1340
        GVSN                  : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1340
        Parent UID            : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v45
        Replicated folder     : 7C764107-1521-4263-B827-04B0155C6067
  Total number of inbound updates being processed: 16

  Updates scheduled:
      [1] Update name: NlsLexicons0001.dll
        Path                  :
        UID                   : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1354
        GVSN                  : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1354
        Parent UID            : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v45
        Replicated folder     : 7C764107-1521-4263-B827-04B0155C6067
      [2] Update name: NlsData0c1a.dll
        Path                  :
        UID                   : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1352
        GVSN                  : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1352
        Parent UID            : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v45
        Replicated folder     : 7C764107-1521-4263-B827-04B0155C6067
...
      [19] Update name: NlsLexicons000d.dll
        Path                  :
        UID                   : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1361
        GVSN                  : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1361
        Parent UID            : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v45
        Replicated folder     : 7C764107-1521-4263-B827-04B0155C6067
      [20] Update name: NlsLexicons000f.dll
        Path                  :
        UID                   : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1362
        GVSN                  : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v1362
        Parent UID            : {6FF5D3E2-7B1C-4551-915E-8FA73BC713BC}-v45
        Replicated folder     : 7C764107-1521-4263-B827-04B0155C6067
  Total number of inbound updates scheduled: 20
```

Source: Microsoft TechNet

Overall, the logging is robust and even allows for detail tracing to be turned on, but it cannot be called user friendly and is more appropriate for use by a developer than a network admin. This is common with trace systems as they are written by developers for their own use and the trace logging in DFSR is no exception.

## Technology Choices

Fortunately, there are replication technology options depending on what you are trying to accomplish. Microsoft has been building basic replication into the server version of Windows for many years, but Windows Server 2003 was the first version to have a more sophisticated implementation that was suitable for usage over a WAN for purposes other than replication needed between Domain Controllers. There has been a steady evolution since Windows Server 2003 R2 and most recently Windows Server 2008 and 2008 R2 supporting slightly better implementations, but each iteration is bound to a particular version of Windows Server. Conversely, there exists replication technologies such as PeerSync which offer everything provided by DFSR and is able to support the latest features on all Windows servers, something that DFSR does not allow.

Features like data compression, byte level replication and multi-threading are key requirements for enterprise-class replication technology.

## PeerSync - A Best-of-Breed Replication Technology

Developed by Peer Software, PeerSync offers a high performance yet flexible replication technology designed to securely replicate and synchronize gigabytes and even terabytes of data across a WAN. PeerSync is a powerful tool that can be used to backup data over the WAN or to enable real-time, bidirectional synchronization when used in concert with PeerLock to avoid file version conflicts.

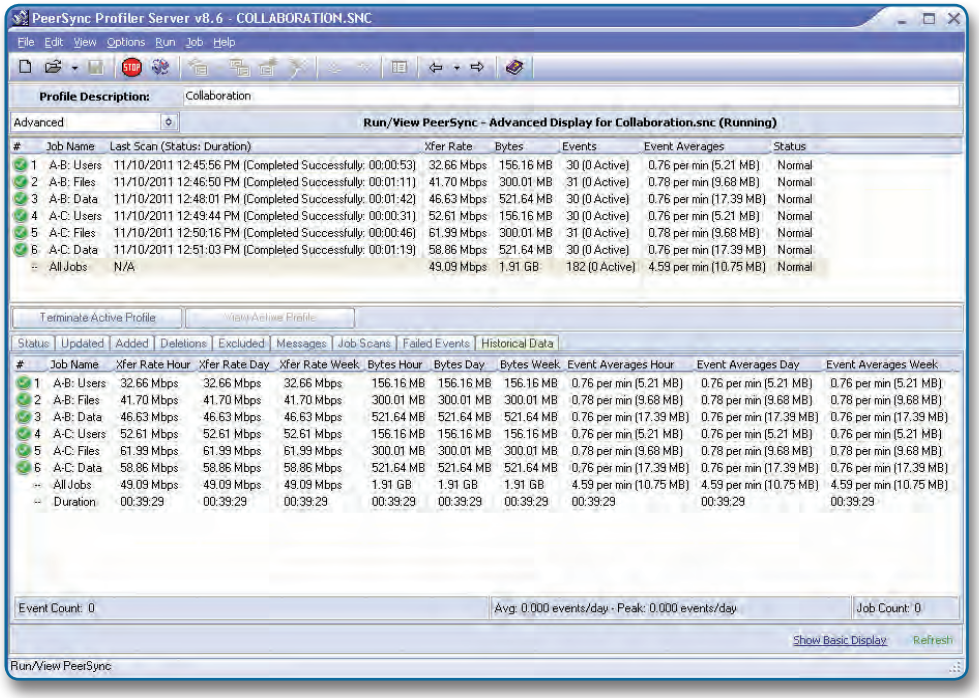
With PeerSync, you have the same full functionality on all of your Windows Servers and you can configure scheduled synchronization jobs, which can pull or push data to remote servers where you cannot install the software. It even supports the ability to synchronize data across different domains.

If an organization needs to synchronize a number of branch locations with a central server, features like data compression, byte-level replication, and multi-threading are critical. While multi-threading is important for scalable performance, it is important how it is implemented. PeerSync leverages several different thread pools for maximum performance. There are sync threads, real-time event threads, and scanning threads, which make each of these areas scalable.

In a backup scenario where 15 branch servers point to a central HQ server, that central server needs a lot more power to handle the traffic/requests for the connected branch server. With a limitation of 16 threads available in DFSR, this server can get into serious problems queuing up sync requests, especially with numerous changes in large files, the central server will be very busy executing block level comparisons at the file level.

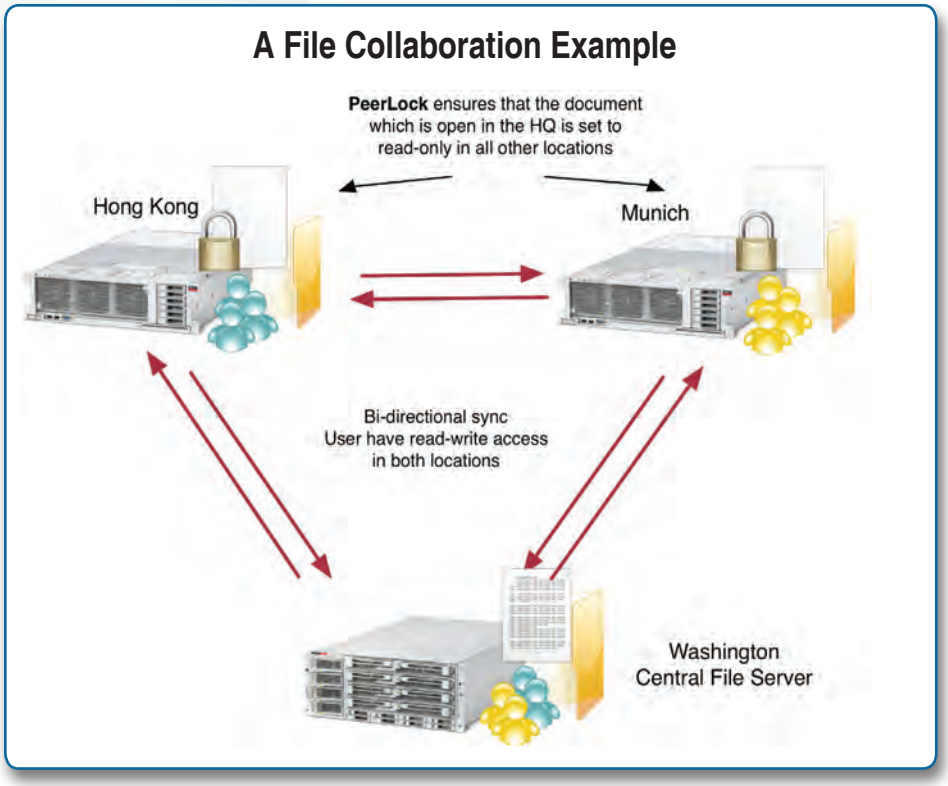
With PeerSync you can configure the threads per job so that the more time critical jobs which need to move larger amounts of data can get more threads, as shown in the screenshot below. PeerSync can run with 60+ threads in the HQ server to quickly manage the incoming requests.

Unlike DFSR, PeerSync is fully integrated with PeerLock so that PeerLock can leverage the jobs configuration in PeerSync. Comprehensive monitoring is also available in PeerSync, providing historical data to help you manage job performance.



## PeerLock Completes the Solution

Depending on bandwidth and the amount of data to be replicated, the lack of locking functionality in DFSR will increase the risk of version conflicts. To address this risk, Peer Software's PeerLock provides version conflict prevention with distributed file locking. As soon as a file is opened for WRITE access, PeerLock notifies the remote servers to lock the corresponding remote copies so users can only open them in READ-ONLY mode while the source copy is in




use. In a multi-master environment where you replicate files bidirectionally between servers, and users need to work with the files on all sites, file locking is a must have capability, not just a nice to have feature.

PeerLock is a server-side software and requires no installation on the client. It works with all file types that maintain a file handle on a single server. An example of a non-supported file type is .txt files. PeerLock can work with both DFSR and PeerSync. The main difference is that PeerLock has to release a file after the changes are made and the file is closed so that the replication software can push the changes to the target. DFSR will not lock the file on the target system while it is queued and waiting for replication. PeerSync, on the other hand, will re-apply the file lock and keep it until replication is complete.

## Feature Comparison

The following table gives you an overview of the different features in Microsoft's DFS Replication and Peer Software's PeerSync.

| Feature | Win 2003 | Win 2003 R2 | Win 2008 | Win 2008 R2 | PeerSync |
|---|---|---|---|---|---|
| Differentials | ● | ● | ● | ● | ● |
| Remote Differential Compression (RDC) | | | ● | ● | |
| Byte Level Replication (RDC alternative) | | | | | ● |
| Open File Support (VSS) | | | ● | ● | ● |
| Multi Threading (Copy Jobs) | | 4 | 16 | 16 | 60+ |
| Multi Threading (Scan Jobs) | | | | | 8+ |
| eMail Reporting | | | | | ● |
| MMC Snap In | ● | ● | ● | ● | |
| Monitoring Console | | | | | ● |
| Bandwidth Throttling | | | ● | ● | ● |
| TCP Transfer | | | | | ● |
| FTP Transfer | | | | | ● |
| File Revisioning | | | | | ● |
| File Locking during Transfer | | | | | ● |
| Blackout Settings | | | | | ● |
| Support for long File Names (256+ characters) | | | | | ● |
| Auto File Recovery | | | | | ● |
| Data Compression | | | ● | ● | ● |
| Remote Connections to NAS Devices | | | | | ● |
| Support for Cross Domain Sync | | | | | ● |
| Support for NFS to NSF Sync | | | | | ● |
| NetApp CIFS Support for real-time sync | | | | | ● |
| Run as a Service | ● | ● | ● | ● | ● |
| Integration with PeerLock | ●[1] | ●[1] | ●[1] | ●[1] | ● |
| Detailed Log Files | | | ● | ● | ● |
| | | | | | |

●[1] - No File Locking during the File Transfer - Depending on the number of files / data in the queue this increases the risk of version conflicts

## Summary

Replication technology is a key component in nearly every computing enterprise as it is essential to applications such as backup, file distribution, file synchronization as well as file sharing and collaboration. Microsoft originally offered the File Replication Service (FRS) to provide limited functionality to assist with domain management. MS DFSR evolved from FRS, but it still seeks to service Microsoft's needs first and suffers from several shortcomings including server OS version dependence and the lack of a file locking and monitoring capability. Peer Software helps you overcome these shortcomings by rounding out DFSR's capabilities via PeerLock as well as PeerSync, a robust, feature filled replication technology.

As an administrator responsible for a large network you need control and visibility into what is going on in your replication system. PeerSync as a standalone application, or working in concert with PeerLock, delivers industry leading performance, control, and insight to meet your replication needs.