

# DFS Does not Replicate Temporary Files

This will remove the background temp archive attribute

```
Get-childitem "D:\Data" -recurse | ForEach-Object -process {if (($_.attributes -band 0x100) -eq 0x100) {$_.attributes = ($_.attributes -band 0xFEFF)}}
```

There are some attributes that also will stop replication. Below command will remove those for whichever folder and subfolders you run this on.

```
attrib * -r -a /S /D
```

REED-01-SRV02

```
Get-childitem I:\DFS -recurse | ForEach-Object -process {if (($_.attributes -band 0x100) -eq 0x100) {$_.attributes = ($_.attributes -band 0xFEFF)}}
```

RHSC-01-SRV01

```
Get-childitem "D:\NEW DFS" -recurse | ForEach-Object -process {if (($_.attributes -band 0x100) -eq 0x100) {$_.attributes = ($_.attributes -band 0xFEFF)}}
```

RHSC-26-SRV03

```
Get-childitem "D:\WilliamsDFS" -recurse | ForEach-Object -process {if (($_.attributes -band 0x100) -eq 0x100) {$_.attributes = ($_.attributes -band 0xFEFF)}}
```

Script for task manager:

~~~~~BAT file~~~~~

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe  
C:\Accent\Scripts\DFSR\_archive\_temp\_bit\_buster.ps1

~~~~~powershell ps1 file~~~~~

```
Get-childitem I:\DFS -recurse | ForEach-Object -process {if (($_attributes -band 0x100) -eq 0x100)
{$_attributes = ($_attributes -band 0xFEFF)}}
```

~~~~~

Windows Powershell - "Running scripts is disabled on this system"

set-executionpolicy remotesigned

<http://www.faqforge.com/windows/windows-powershell-running-scripts-is-disabled-on-this-system/>

If you don't want it to work against subdirectories just remove the -recurse parameter.

11 Nov 2008 7:40 AM

- [Comments11](#)

Note that this post has been added to the [TechNet Wiki](#) to allow for community editing.

If you notice that DFS Replication (DFSR) is not replicating certain files, one simple reason is that the temporary attribute is set on them. By design, DFSR does not replicate files if they have the temporary attribute set on them, and it cannot be configured to do so.

This may not be obvious because nearly all the normal methods you would use in Windows to check file attributes do not show the temporary attribute. Specifically, all of the following do not show the temporary attribute - [Attrib.exe](#), Explorer's file properties, [FileSystemObject](#) in Windows Scripting Host, and [CIM\\_Datafile](#) in WMI. Also, DFSR does not log any errors to the event log or to the debug logs to show temporary files are not being replicated. There is a relevant entry in the debug logs, but it is not an error because this behavior is by design.

The reason DFSR does not replicate files with the temporary attribute set is that they are considered short-lived files that you would never actually want to replicate. Using the temporary attribute on a file keeps that file in memory and saves on disk I/O. Therefore applications can use it on short-lived files to improve performance.

An application can use FILE\_ATTRIBUTE\_TEMPORARY when calling the [CreateFile](#) function if they want a temporary file. But an even better way is to also specify FILE\_FLAG\_DELETE\_ON\_CLOSE so the temporary file is deleted when all handles are closed. That way you get the performance benefit of a temporary file (it's kept in memory) and it is removed when handles are closed so administrators don't come along and wonder why DFSR isn't replicating it.

If you have temporary files that you want DFSR to replicate, you may think it is enough to just remove the temporary attribute on those files and be on your way. And you can do that. But since you got in this situation once, it is likely you still have an application that will come right back and create more temporary files. So you need to get at the crux of the issue - why do you want to replicate files that an application is specifically creating to be temporary? Either the application must change its behavior, or you must except that temporary files won't be replicated, because there is no way to make DFSR replicate files as long as the temporary attribute is set on them.

Checking the Temporary Attribute on a File using Fsutil

But wait, you say, maybe I don't even know yet if these files that aren't replicating are temporary! So let's find out. As mentioned before, almost none of the ways to check attributes in Windows will actually show the temporary attribute. But there is one that does - the handy [Fsutil](#) tool that is included in Windows.

```
fsutil usn readdata c:\data\test.txt
```

Major Version : 0x2

Minor Version : 0x0

FileRef# : 0x0021000000002350

Parent FileRef# : 0x0003000000005f5e

Usn : 0x000000004d431000

Time Stamp : 0x0000000000000000 12:00:00 AM 1/1/1601

Reason : 0x0

Source Info : 0x0

Security Id : 0x5fb

File Attributes : 0x120

File Name Length : 0x10

File Name Offset : 0x3c

FileName : test.txt

File Attributes is a bitmask that indicates which attributes are set. In the above example, 0x120 indicates the temporary attribute is set because that is 0x100 and 0x20 (Archive) = 0x120.

Here are the possible values:

READONLY	0x1
HIDDEN	0x2
SYSTEM	0x4
DIRECTORY	0x10
ARCHIVE	0x20
DEVICE	0x40
NORMAL	0x80
TEMPORARY	0x100
SPARSE_FILE	0x200
REPARSE_POINT	0x400
COMPRESSED	0x800

OFFLINE	0x1000
NOT_CONTENT_INDEXED	0x2000
ENCRYPTED	0x4000

You combine the values to come up with the File Attributes bitmask value.

If you need a sanity check:

1. Start, Run, Calc.
2. Change to Hex and paste in the File Attributes value from the Fsutil command. Say for example, 4925.
3. Hit the And button, then type 100.
4. Hit equals and if it returns 100, then the temporary attribute is set. If it returns 0, the temporary attribute is not set.

### Checking for Temporary Files in the Debug Logs with Findstr

Another way to check if files are not replicating because they have the temporary attribute set is to use [Findstr](#) (included in Windows) to look for the FILE\_ATTRIBUTE\_TEMPORARY text string in the DFSR debug logs.

First you need to extract out all of the debug logs, because all except the active log will be compressed, as indicated by a .GZ extension. The DFSR debug logs (Dfsr\*.log and Dfsr\*.log.gz) reside by default under %windir%\debug. All the popular compression tools such as Winzip and Winrar can handle .GZ compression.

Let's say you extracted the debug logs to C:\Logs. You can then run the following Findstr command to look for temporary files.

```
Findstr FILE_ATTRIBUTE_TEMPORARY c:\logs\dfsr*.log
```

That will output the entire line for every line in the debug log that contains a match to that string. If it doesn't find any matches, it will return to a prompt and not show anything.

Sample output from a matching entry:

```
C:\WINDOWS\debug\Dfsr00018.log:20080903 16:14:29.390 1808 USNC 1204
UsnConsumer::ProcessUsnRecord Skipping USN_RECORD with FILE_ATTRIBUTE_TEMPORARY flag:
```

If it does find any matches, you can then open the specified log file, search on the string FILE\_ATTRIBUTE\_TEMPORARY (Ctrl+F or Edit | Find in Notepad) and then you will see the actual file name for the file that was skipped because the temporary attribute is set on it.

### Removing the Temporary Attribute from Multiple Files with Powershell

So you figured out that DFSR is not replicating some files because they have the temporary attribute set. There is no way to change this behavior in DFSR, so the only option is to live with it, or remove the temporary attribute from the files you want to replicate. An application in your environment has created these temporary files, so just treating the symptom isn't enough, you need to find the application that creates them and either change its behavior, or accept that those files will not be replicated.

Since Attrib is not aware of the temporary attribute, we need to go to greater lengths to remove it. First you need to have Powershell installed on the machine - [www.microsoft.com/powershell](http://www.microsoft.com/powershell)

Then bring up a Powershell prompt (Start, Run, Powershell or from the Programs menu) and run this command to remove the temporary attribute from all files in the specified directory, including subdirectories (in this example, D:\Data):

```
Get-childitem D:\Data -recurse | ForEach-Object -process {if ((($_.attributes -band 0x100) -eq 0x100) {$_attributes = ($_attributes -band 0xFEFF)}}
```

If you don't want it to work against subdirectories just remove the -recurse parameter.

Pasted from <<http://blogs.technet.com/b/askds/archive/2008/11/11/dfsr-does-not-replicate-temporary-files.aspx>>

---

Revision #1

Created 23 December 2023 04:30:20 by ColtM

Updated 13 June 2024 01:28:01 by ColtM