During the initial troubleshooting of Files Connect crashes, high CPU usage cases, slow performance cases, connectivity and access permission issues, we focus on two main areas during the troubleshooting: file system health and permissions configuration. The issues covered in this article account for the majority of support cases, so they are very important to address.

File System Maintenance

As you may know, Files Connect is a user mode application. It does not install any kernel drivers or file system filters. This means that Files Connect calls standard Windows APIs and waits for the system to return the result of the call An advantage of this is while the application itself might stall or crash, it can't bring down the server. As a result of this architecture, when troubleshooting cases involving data stored in Windows hosted NTFS volumes, it's important to address any possible filesystem issues, whether it's a corrupted Windows NTFS volume, an overly fragmented Master File Table. For customers storing data on non-Windows hosted volumes, such as NAS devices, and using the Files Connect Network reshare feature, please refer to the device vendor for any guidance on filesystem integrity featured and skip to the section below on optimizing permissions.

Note: For Windows Server 2008 R2, Microsoft has identified an issue in NTFS that can cause the server to become unresponsive. Our article at https://kb.acronis.com/content/39356 explains what the issue is, some ways that we have been able to recreate the filesystem issue in our test lab and potential workarounds to prevent the issue from happening. This issue has been fixed in Windows Server 2012 and higher.

First, we recommend checking any Windows NTFS filesystem using the chkdsk /f command on each drive. Since chkdsk in read-only mode may not detect all corruption on Windows NTFS volumes, it's important to use the /f switch with chkdsk to find and address all issues, but this requires read/write access. As a result, it will take the volume offline, so this needs to be done during a maintenance window. We want to confirm a healthy filesystem before the steps that follow, in an effort to minimize any chance of triggering an outage. To be clear, the /f switch is only concerned with the logical filesystem, we don't recommend any options that perform a surface scan, since this is known to consume inordinate amounts of time and serves no purpose on modern RAID systems. Microsoft has dramatically improved runtimes for chkdsk /f and our case histories show typical runtimes on Windows Server 2008 R2 of 20-40 minutes, faster on newer OS versions. Runtimes are impacted by the total number of objects (files and folders) in the filesystem and the amount of available RAM in the system. The size of the data itself is not a factor.

After that completes successfully, the next step is to examine fragmentation of the Master File Table (MFT) on each volume. The MFT is a catalog in the logical filesystem that Windows uses to keep track of where all of the files are in the logical filesystem, independent of the underlying hardware. Additional fragments require additional IO calls, so fragmentation of the MFT will affect the performance of user mode applications like Files Connect. This is true for both physical and virtual deployments. To check this, you request a verbose disk fragmentation analysis report for each disk. This command with the /a switch will only perform an analysis of the drive, it won't make any changes. The /v switch requests the

verbose version of the report. To run the command and pipe the output to a text file for review, from a command prompt, type something like:

defrag /a /v X: > C:\X_Drive.txt

Repeat the command, changing X to each drive letter to be analyzed. The verbose version of this report contains a line detailing the state of the Master File Table (MFT) that we are concerned with, we expect it to be in the single digits.

It's not unusual for performance cases to have MFT fragments in double digits. If that's the case, the recommended next step is to defragment the volume. The Microsoft defrag tool requires a minimum of 15% free space on a volume in order to defrag the MFT. We are aware there are third party defrag tools, but we don't have sufficient case history to recommend any. Once this maintenance is completed, it would be best to run the verbose analysis again, to confirm the state of the MFT. We are looking for low single digits.

After all the necessary disk maintenance steps have been completed and we have a confirmation of healthy filesystems, we move on to permission configurations.

File System Permissions

When we examine the permission configuration we address two main areas:

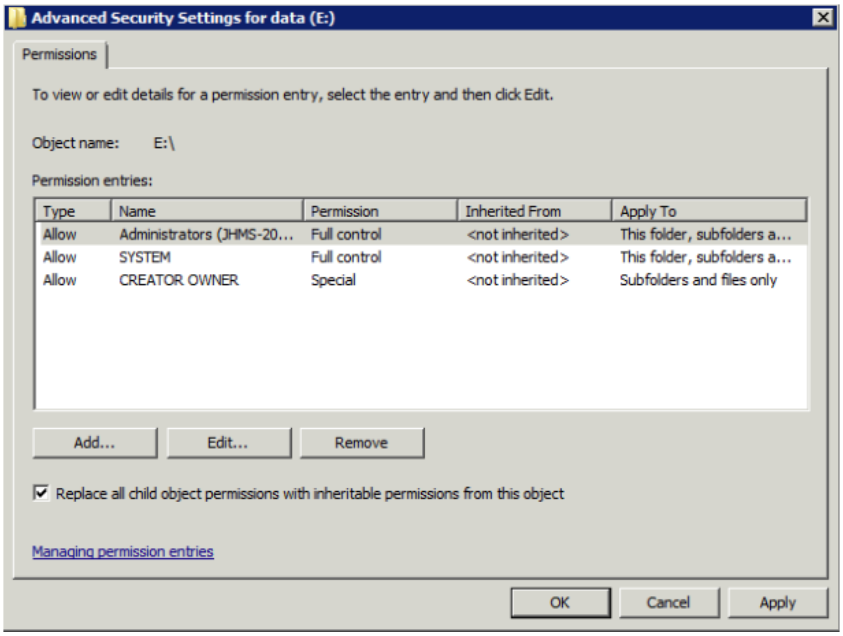
- The Files Connect service account and its permissions.
- User account permissions.

The Files Connect service account and its permissions

Files Connect requires an explicit ACE, not as a member of a group, for the service account, which defaults to the local SYSTEM account, to have NTFS Full Control over entire drives or remote shares. On local drives, another requirement is for the service account to inherit its permissions from the drive letter, so the service account ACE is propagating unbroken down the hierarchy. Our case histories show that when inheritance is broken for the Files Connect service account, it can result in intermittent access/permission issues that are more likely to manifest when the server is operating under load. We believe this is due to additional overhead delaying resolving permissions for the service account, which is used continuously. Based on our case histories, the most reliable and effective permission configuration at the root of a drive is also the most streamlined, as follows:

Administrators, Full Control, Applies to this folder, subfolders, and files
SYSTEM, Full Control, Applies to this folder, subfolders, and files
CREATOR OWNER, Modify+ [more on this below], Applies to subfolders and files only

A note regarding CREATOR OWNER: since it applies to subfolders and files only, Full Control will display as "Special" on Windows Server 2008 and above.



User account permissions

A review of Files Connect case histories have revealed widespread issues with the way permissions are managed that result in performance and functional issues. The primary way Files Connect handles user account permissions resolution is by letting Windows do it. A command request comes from a Mac client via AFP to the Files Connect service. Then Files Connect converts that command into Windows API calls and send them to Windows with the security token of the user account used to connect to the server from the Mac. This way the security is always enforced for those user accounts. We need to make sure we streamline the resolution of those permissions as much as possible. You paid a lot of money for your server and you don't want it to waste cycles doing something it doesn't need to be doing, or do it in an inefficient way. Additionally, if it returns a value to the Mac and the Mac doesn't understand that value, it may fail or it might ask again. And as mentioned, we tend to see the correlation of issues happening more when the server is under load. Often when we get on remote sessions during a slow time because this is usually the time customers can meet, everything just works, and the customer can't reproduce the issue.

As it turns out, there isn't a predefined Windows permission bundle short of full control that gives Mac clients everything they need to work with files on the server. In Windows, there are the predefined permissions bundles Read, Execute, Modify, and Full Control. These are all bundles of the underlying granular NTFS permissions. Macs understand Read & Execute as Read Only and have no issue working with that particular permission bundle.

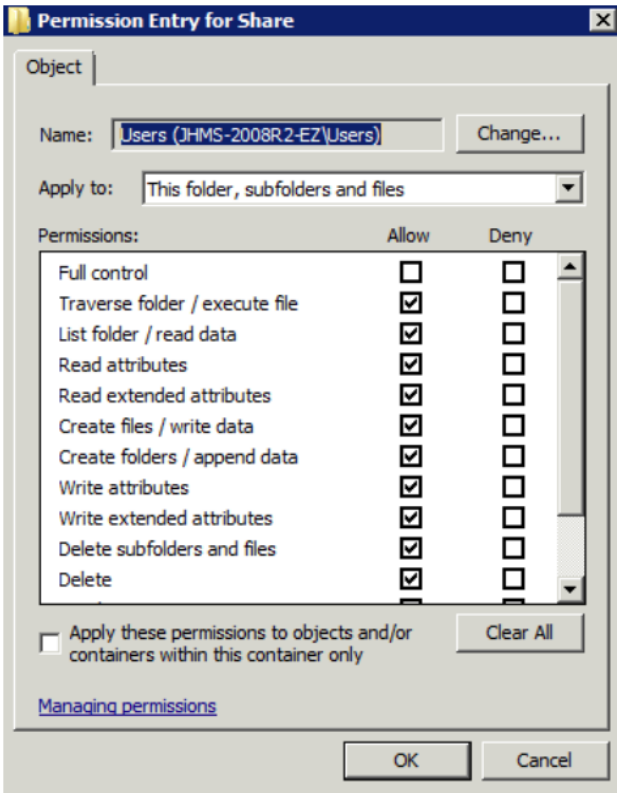
We don't actually recommend Full Control, since there is a possibility that if a Mac client inadvertently connects to the server via SMB, they could modify the permissions on files and folders and create problems where there wasn't one. By default, Files Connect silently drops these requests from Mac clients to modify permissions. There is a Files Connect setting 'Allow Mac client to change permissions'

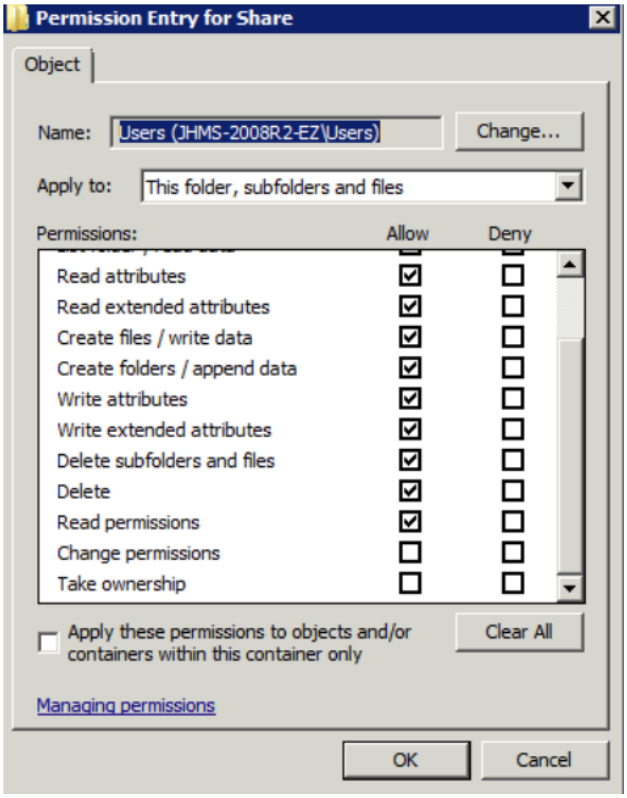
that’s off by default and we don’t recommend enabling it and it only applies when Macs connect via AFP. When they inadvertently connect via SMB and they have Full Control, Mac apps frequently will request to modify the permissions on a file they are working on, and typically that modified permission is undesirable, because what it typically does is strips off all the other permissions except for the person working on the file. If you ever had a situation, where people say, “Someone else was working on a file and now I can’t open it” this could very well be a consequence of that issue. For this reason, we generally don’t recommend for Full Control to be granted. Macs don’t need the 2 granular NTFS permissions of ‘Take ownership’ and ‘Change permissions’.

Most user account permission problems are related to the Windows Modify permission bundle. The issue is the Modify bundle doesn’t have all the required permissions for Mac clients to work on files on a server share. This permission bundle works well for Windows clients, but Macs are not happy with it and unfortunately they don’t indicate this in any clear way and many times it will seem to work for a while. We suspect that it fails mostly when the server is under load.

Having said all that, it mostly comes down to an issue where Macs need more than the Windows Modify bundle to work with the files on a share, specifically the NTFS permission ‘Delete subfolders and files’. Windows clients don’t need that permissions if the user account has an explicit Delete permission on the object. However, when Mac clients are browsing using Finder, the Mac frequently will send a request ‘can I delete child objects in this folder?’ and if the user account hasn’t been granted ‘Delete subfolders and files’ sometimes the server will say ‘no’, even if those objects have Delete permission configured on them, which is intermittently misunderstood as the folder being read only. So, the solution is to upgrade the Modify bundle to include ‘Delete subfolders and files’. For the sake of brevity, we refer to this as “Modify+”.

To summarize, for typical workflows, the Mac users will need all the granular NTFS permissions except Full control, Change permissions, and Take ownership.

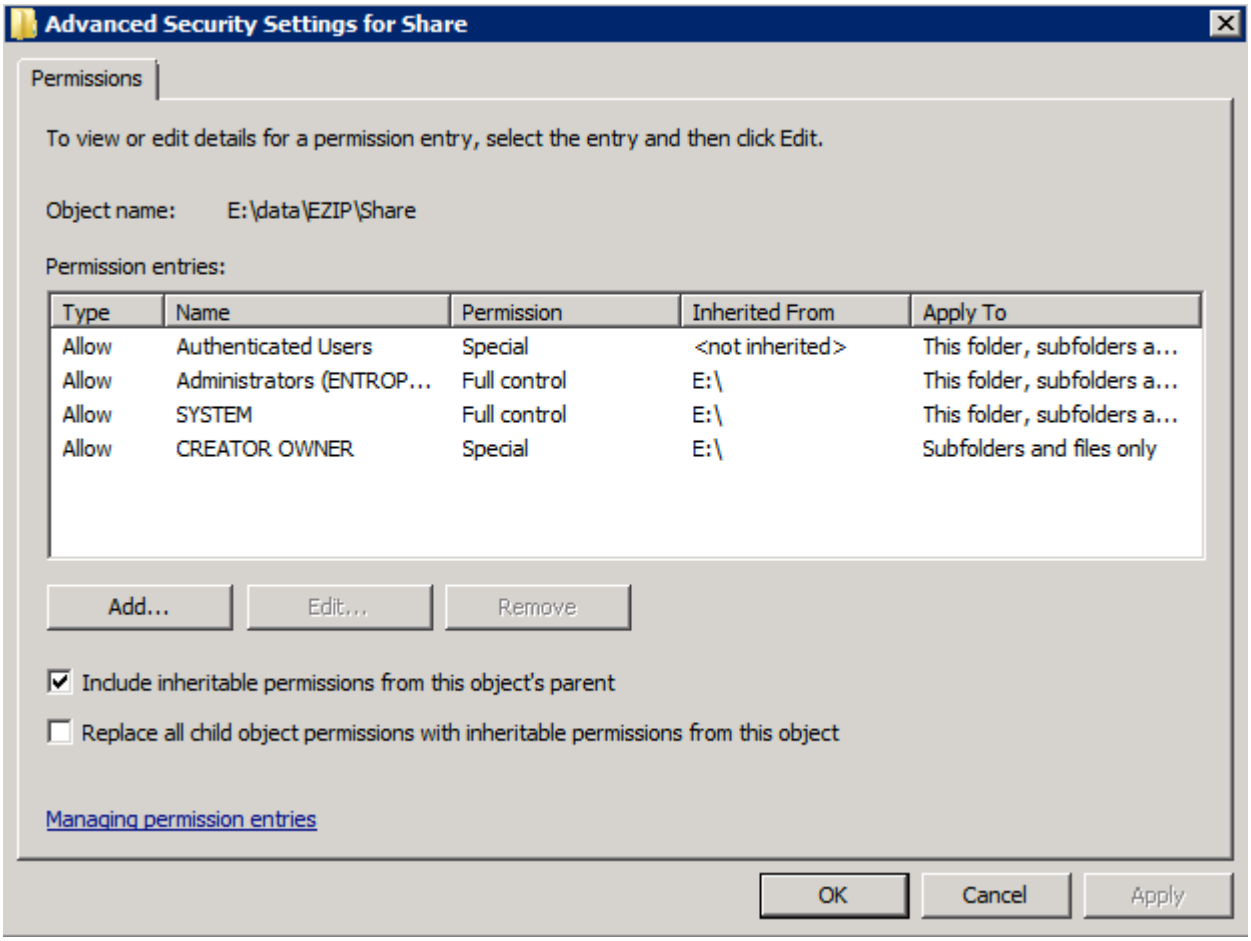




Listed out, this is:

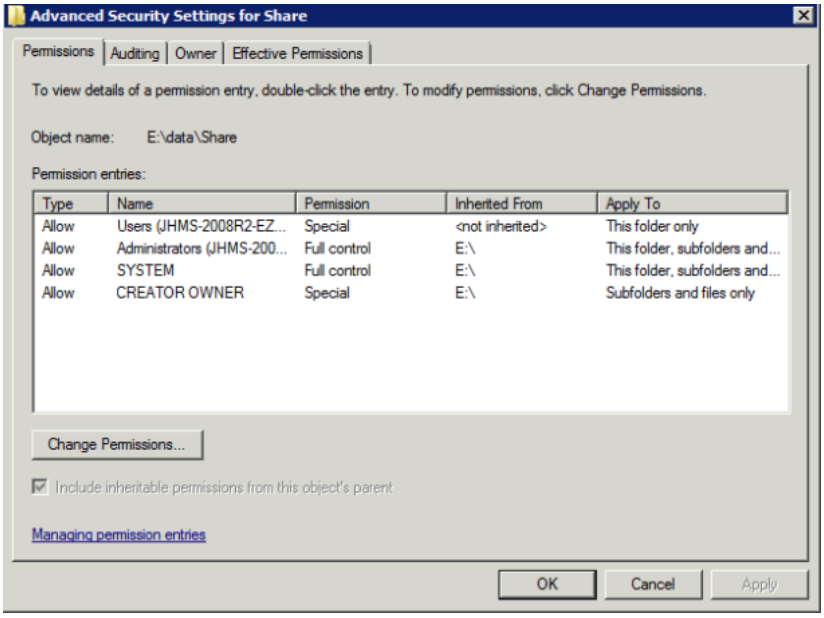
- Traverse folder/execute file
- List folder/read data
- Read attributes
- Read extended attributes
- Create files/write data
- Create folders/append data
- Write attributes
- Write extended attributes
- Delete subfolders and files
- Delete
- Read permissions

A basic share would look something like:



The advantage of using inherited permissions in this fashion is that each time you create a subfolder, Windows automatically applies the proper permissions to it, using the inheritance settings the server administrator defined. Without these settings, the administrator would be forced to define permissions from scratch for each new subfolder. That can be a lot of extra work, with the potential for errors and inconsistencies. More importantly, if you decide to change the permissions later-for instance, changing the Full Control permission for subfolders from the ‘Everyone’ group to a more limited group of users-you can make a single change and have the changes apply to all the child folders automatically.

Now, if you need restricted permissions at the root of a share, generally the best advice we give is to apply an explicit ACE with limited permissions, such as read only, for a staff group to a parent folder, typically the directory that is being shared, and change the "Apply to" setting to "This folder only" so Mac clients can mount the share. That way the staff group ACE will not inherit down but the SYSTEM ACE will.



Then on the child folders, apply an explicit ACE granting permissions to each particular group. Don't forget that the .TemporaryItems at the root of all shares would need an explicit ACE granting Modify+ to all user accounts that might use a Mac client, so we suggest a broad group such as “Authenticated Users”. This will support the safe save operation documented in the Apple KB article at: http://support.apple.com/kb/TS3752

You can see on the screenshot below that the child folder "test" did not inherit the read only ACE from the parent folder "Share" pictured in the screenshot above. Then we applied an explicit ACE on "test" granting full control the group "Users".