NTFS Tutorial

Day 4:

Daily Operations

Carsten Schaefer

NTFS Permissions vs. Share Permissions

Difference between NTFS Permissions and Share Permissions

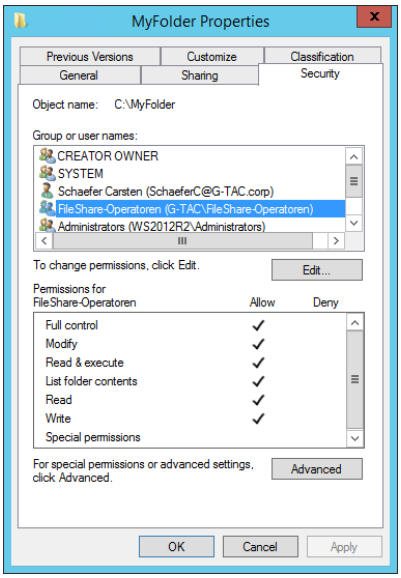
Share permissions are applied when a shared folder is accessed over a network. It is a common misconception to think that share permissions works in a different way. When you log in locally to a Windows machine (even if a file or folder is shared to other users within your network), every time you access an object locally, NTFS permissions apply and share permissions do not apply. It does not matter how restrictive share permissions have been set up on your network, if you have access to the object and you are logged into the workstation or server that “owns” the file or folder, you will be granted access.

Combining NTFS Permissions and Share Permissions

When using share permissions and folder permissions please keep in mind, that you can apply different NTFS permissions to each folder within a shared folder. Working this way will ensure a permission strategy for each kind of data located in an appropriate folder structure.

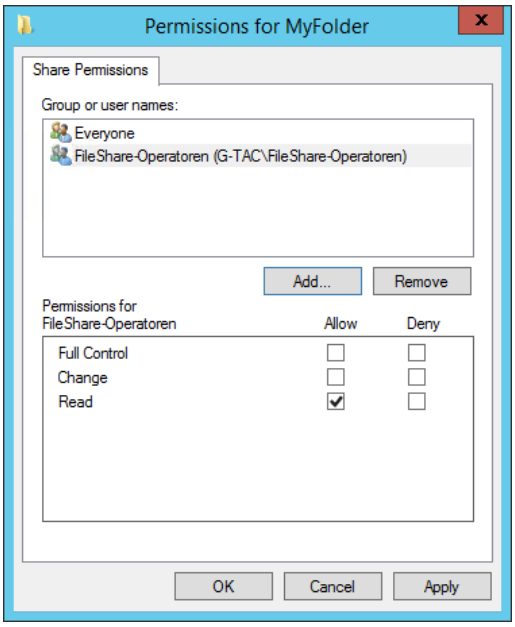
A frequently asked question when managing Windows Server environments is: once you combine share permissions with NTFS permissions, how do these two types of permissions work together? The answer is rather simple and helps you determine the most effective form of permission for a shared folder. Both sets of permissions get applied and the more restrictive of the two takes precedent. To give you a better idea, take a look at the below example.

You give “Full Control” NTFS permissions to the “FileShare-Operatoren” group for a folder called MyFolder, as seen in the image below:



Full Control permissions for MyFolder

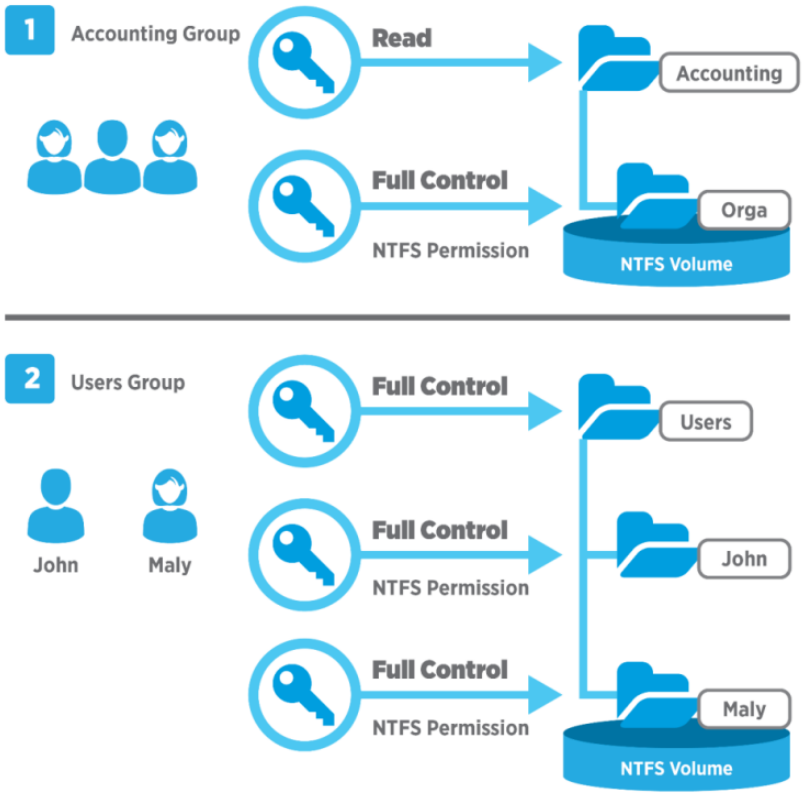
If you share MyFolder within the Windows Network to the “FileShare-Operatoren” group using “Read” permissions and a user that belongs to this group tries to access the folder from the network, that user will only have “Read” access and not “Full Control”. However, if that user then goes to the workstation or server where MyFolder is allocated, he will be granted “Full Control” permissions.



Read Only share permissions for MyFolder

3 Examples of Combining Share Permissions with Folder Permissions

In the next two examples we have shared folders on NTFS volumes. These shared folders contain subfolders that have also been assigned NTFS permissions.

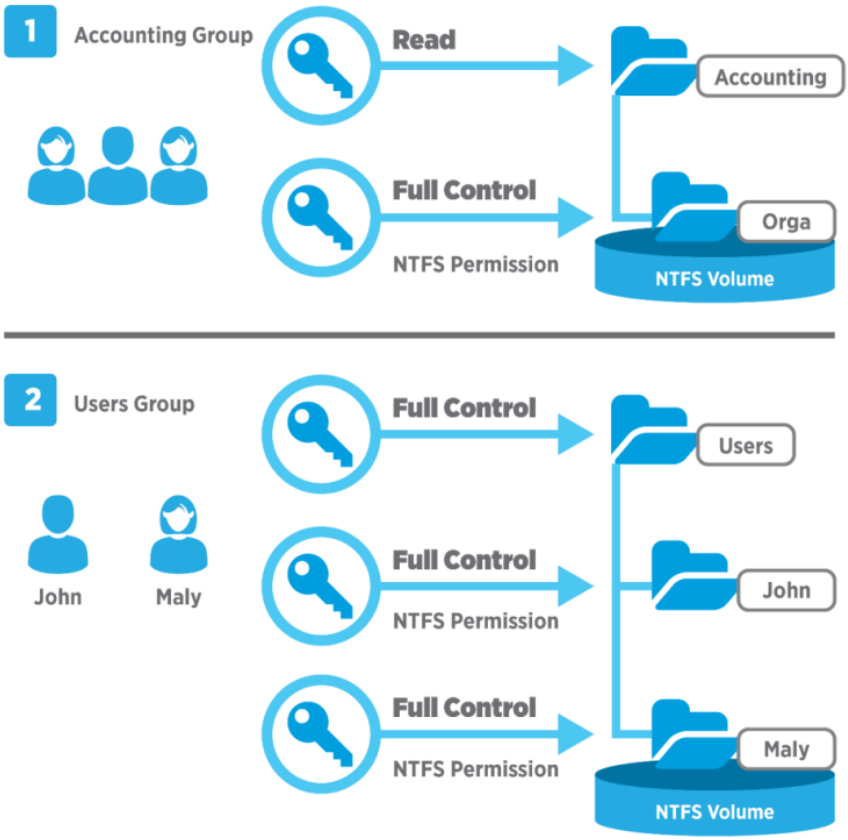


Combined Share and NTFS Permissions

First example:

- Accounting folder is shared.
- The Accounting group has the shared folder “Read” permission for this folder and the NTFS “Full Control” permission for the Orga subfolder.

The **effective permissions** for any member of the Accounting group for the subfolder called Orga is Read.



Combined Share and NTFS Permissions

Second example:

- Users folder contains home folders for each user, here John and Maly.
- Both home folders contains data accessible only to the user for whom the folder is named.
- The Users folder has been shared and the Users group has "Full Control" permission for the Users folder.
- John and Maly have the NTFS "Full Control" permission for their home folder only and no NTFS permissions for other folders.
- Boths are members of the Users group.

The **effective permissions** for John and Maly for their own home folder are Full Control. But John has no access to Maly's home folder and Maly has no access to John's home folder.

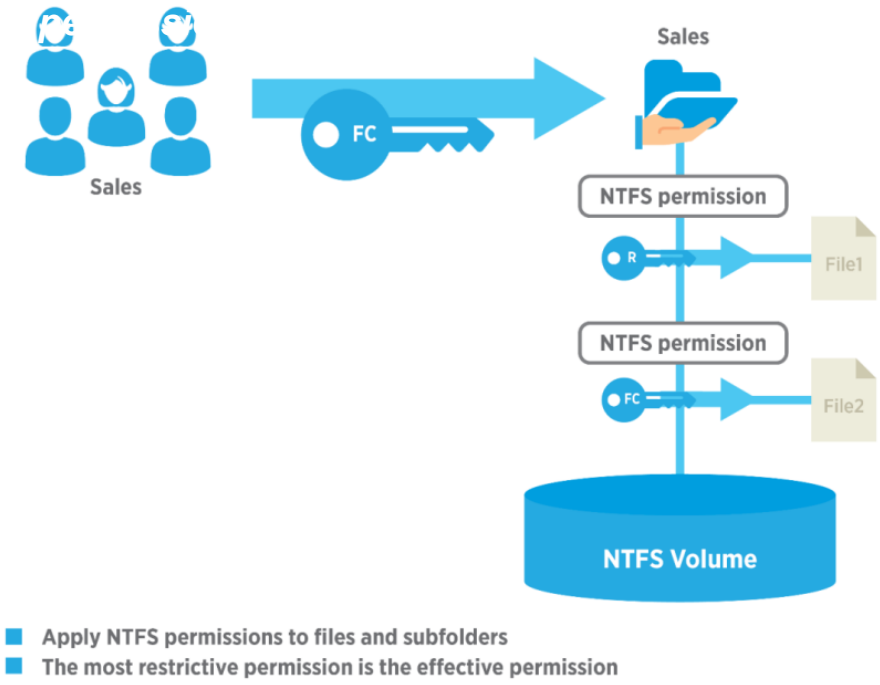
Third example:

The group Sales has these permissions:

- NTFS Permissions Full Control for shared folder Sales
- NTFS Permissions Read for File1
- NTFS Permissions Full Control for File 2

The **effective permissions** are:

- The member of this group are granted only Read access to File1 because it is the most restrictive permission.
- And they are granted Full Control to File2 because both permission assignments are at the same level.



Effective NTFS Permissions

Copying and Moving Files and Folders

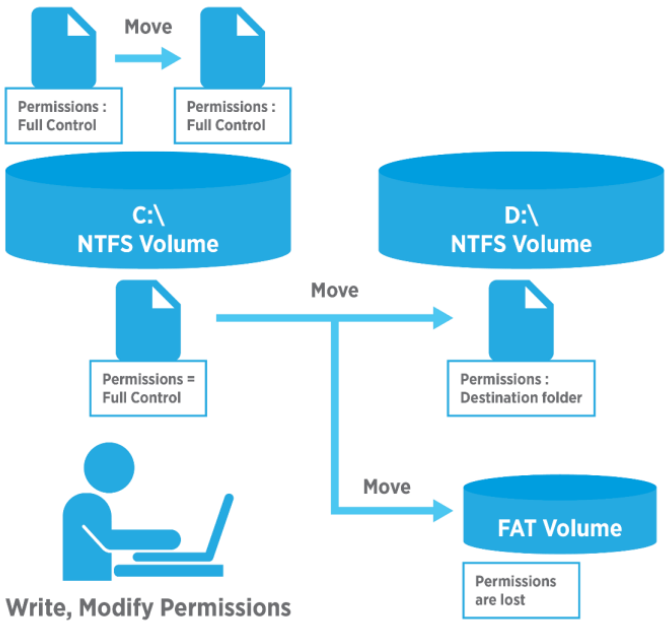
Now that we know how to manage permissions for folders or files, it's a good time to ask ourselves a question: What happens to permissions if I copy or move the files or folders? The answer is: it depends.

To give you a clearer explanation, consider the following three scenarios. Let's assume that you are going to copy "D:\MyFolder" and let's assume that "D" has an NTFS format.

Scenario A: Copy D:\MyFolder to E:\ (E:\ is a FAT volume)

Results:

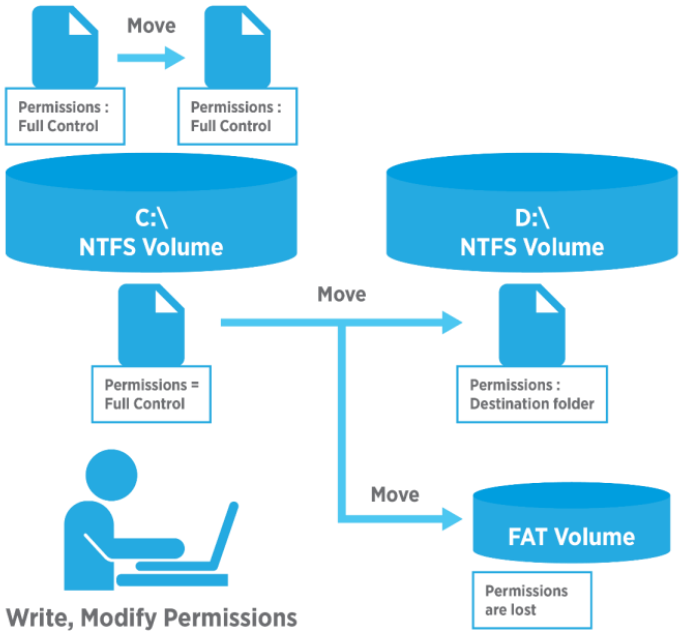
- When you copy files or folders to FAT volumes, the folders and files lose their NTFS permissions because FAT volumes don't support NTFS permissions.



Scenario B: Move D:\MyFolder to D:\MyFiles

Results:

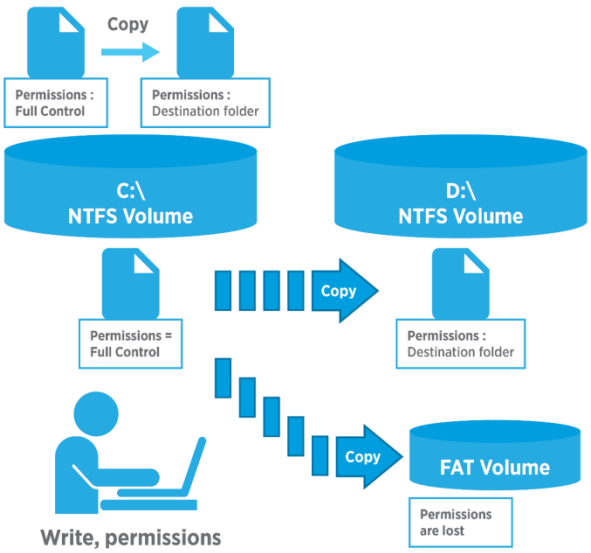
- The file or folder retains its original permissions.
- You must have the “Write” permission set up for the destination folder to move files and folders into that folder.
- You must have the “Modify” permission set up for the source file or folder. The “Modify” permission is required to move a file or folder because Windows 2000 deletes files and folders from the source folder after they are copied to the destination folder.
- You become the creator and owner.



Scenario C: Copy D:\MyFolder to F:\MyFolder (F:\ is an NTFS volume)

Results:

- The file or folder inherits the permissions of the destination folder.
- You must have the "Write" permission set up for the destination folder to move files and folders into that folder.
- You must have the "Modify" permission set up for the source file or folder. The "Modify" permission is required to move a file or folder because Windows XP Professional deletes files and folder from the source folder after they are copied to the destination folder.
- You become the creator and owner.



Powershell and NTFS Permissions

Automation and scripting become more important for system administrators with every Windows Server version. One of the key concepts behind Windows Server 2012 is the capability to do almost everything you can do on a GUI. Using Powershell scripts and NTFS permissions are no exception.

The following table lists the most common use case scenarios that a system administrator can have and how to perform operations such as using scripts without going to the common graphical user interface (GUI). This is just to get you started on managing ACL's with Windows Powershell. Remember, these scripts can become as complicated as you want them to be.

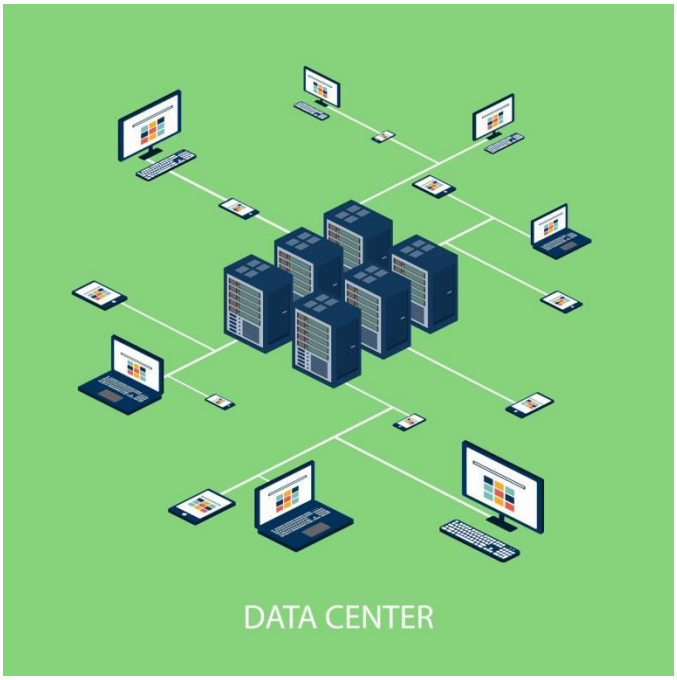
Task	*Powershell Script*
Reading Permissions of a Single File or Folder	((Get-Item D:\MyFolder).GetAccessControl('Access')).Access

Modifying User Permissions on a Folder	$HomeFolders = Get-ChildItem C:Homefolders -Directory
	foreach ($HomeFolder in $HomeFolders) {
	$Path = $HomeFolder.FullName
	$Acl = (Get-Item $Path).GetAccessControl('Access')
	$Username = $HomeFolder.Name
	$Ar = New-Object System.Security.AccessControl.FileSystemAccessRule($Username, 'Modify', 'ContainerInherit,ObjectInherit', 'None', 'Allow')
	$Acl.SetAccessRule($Ar)
	Set-Acl -path $Path -AclObject $Acl
	}

Basic Powershell scripts to manage NTFS permissions

Best Practices

- A common practice by many businesses is to share a folder by giving full access to a group made up of everyone, then control who can access that folder using NTFS permissions.
- Always try to share folders with groups instead of individual people, as this makes administration tasks far simpler.
- To consolidate administration and group files into application, data, and home folders, centralize all home and public folders separately from your applications and operating system. Doing so provides the following benefits: a) permissions may only be assigned to folders, not individual files and b) backing up will be less complex because you will not need to back up application files, as all home and public folders will be consolidated in one location.
- When you assign permissions for working with data or application folders, assign the “Read & Execute” permission to the Users group and Administrators group. This will prevent application files from being accidentally deleted or damaged by users or viruses.
- Always assign the most restrictive permissions that still allow users to perform required tasks. For example, if users only need to read information in a folder and should never delete or create files, assign the “Read” permission.
- Organize your resources so that folders with the same security requirements are located within one folder. For example, if users require “Read” permission for several application folders, store those folders within a single folder. This will allow you to share that larger folder instead of sharing each individual application folder.
- Use intuitive share names so that users can easily recognize and locate resources. For example, for the Application folder, use “Apps” as the share name. You should only use share names that can be used across all client operating systems.


DATA CENTER