NTFS Tutorial

Day 1:

Understanding NTFS Permissions

Carsten Schaefer

Basic Concepts

NTFS

NTFS, which stands for **New Technology File System,** is Microsoft's current file system for the Windows NT operating system. NTFS is the successor of Microsoft's previous systems, FAT and HPFS, and contains a wide range of improvements in terms of performance, extendibility, and security.

The main differences between NTFS and its predecessors are:

- FAT32 only supports individual files of up to 4GB in size. On the other hand, NTFS supports files of up to 16 EiB (16 × 1024^6 or 2^{64} bytes).
- The most important difference you need to understand in order to follow this tutorial is that NTFS supports file permissions and introduced the concept of the **access control list** (ACL), a concept we will be explaining in more detail as we proceed.

NTFS Permissions

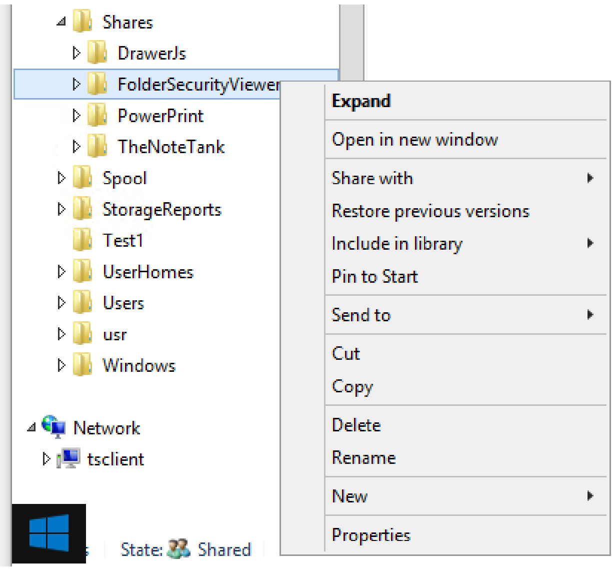
NTFS permissions determine who have access to files or folders. These permissions can be assigned to individual users or groups, but the best practice is to assign them to groups whenever possible. Permissions are set in the ACL.

Access Control List (ACL)

The **access control list** (ACL) is the list of users or groups that have access to a certain object. An object can be a file or folder. Each entry in the ACL is known as an **access control entry** (ACE).

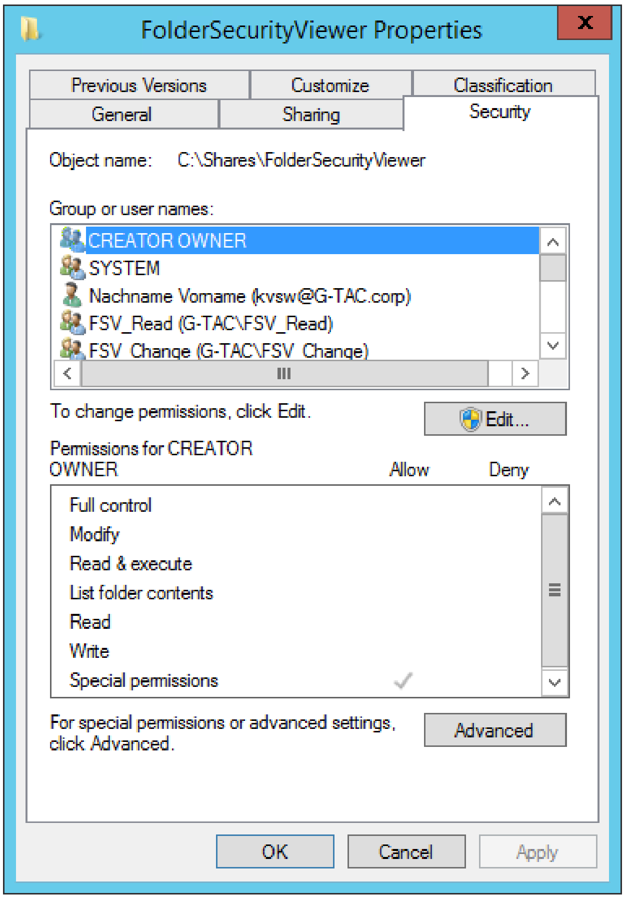
The users or groups in the ACL are known as **trustees**. Permissions can be allowed, denied, or audited.

To create, edit, or view access control lists, you right click on a file or folder then select "Properties" from the options displayed:



Menu options relating to a file or folder in Windows

Next, click on the "Security" tab to display the access control list (ACL) for the chosen file or folder.



Object name: C:\Shares\FolderSecurityViewer

Group or user names:

- CREATOR OWNER
- SYSTEM
- Nachname Vorname (kvsw@G-TAC.corp)
- FSV_Read (G-TAC\FSV_Read)
- FSV_Change (G-TAC\FSV_Change)

To change permissions, click Edit.

Permissions for CREATOR OWNER	Allow	Deny
Full control		
Modify		
Read & execute		
List folder contents		
Read		
Write		
Special permissions	✓	

For special permissions or advanced settings, click Advanced.

An access control list (ACL) in the Windows Server 2012 R2 / 2016

Understanding Permissions

Windows allows you to assign different types of permissions to an object. You can allow or deny such permissions. The types of permissions change depending on if you are working with a file or folder.

NTFS Folder Permissions

You can assign permissions to a user or group for a specific folder and, thus, control their access level. How these permissions are propagated to subfolders and their respective files is controlled by **inheritance**, a concept we will explain in more detail as we proceed. Next table lists and describes all permissions that can be allowed or denied for a certain user or group.

Permission	*Description*
Full Control	Specifies whether a user or group has all available permissions for a folder.
Modify	Specifies whether a user or group can modify the contents of a folder. It is more restrictive than full control, as it does not allow users/groups to change permissions or take ownership of said folder.
Read and Execute	Specifies whether a user or group can read the data within a folder and execute the programs said folder contains.
List Folder Contents	Specifies whether a user or group can list the content of a folder. This does not allow users/groups to run any of the programs or read any of the data within the folder.
Read	Specifies whether a user or group can read the data within a folder. As opposed to "Read and Execute", if there is an executable file within the folder, the user or group will be unable to run it.
Write	Specifies whether a user or group can create files and folders, write data, and write attributes for a folder. The write permission implies the ability to read all data within the folder.
Special Permissions	Refer to TABLE 3 for the list and description of special permissions.

List of NTFS folder permissions

NTFS File Permissions

You can assign permissions to a user or group for a specific file and, thus, control their access level. Next table lists and describes all permissions that can be allowed or denied for specific users or groups. NTFS file permissions take priority over NTFS folder permissions.

For example,

if you have access to a folder, but an administrator denies access for a file within that folder, you cannot access that file even if you have the necessary permissions for its parent folder.

Permission	*Description*
Full Control	Specifies whether a user or group has all available permissions for a file.
Modify	Specifies whether a user or group can modify a file. It is more restrictive than full control, as it does not allow users/groups to change permissions or take ownership of said file.
Read and Execute	Specifies whether a user or group can read the contents of a file and execute the programs of said file.
Read	Specifies whether a user can read a file's data. As opposed to "Read and Execute", if the file in question is an executable file, the user or group will be unable to run it.
Write	Specifies whether a user or group can change the content or, in other terms, write data to a file. The write permission implies the ability to read all the data contained in a file.
Special Permissions	Refer to next table for the list and description of special permissions.

List of NTFS file permissions

Permission	*Description*
Traverse Folder/ Execute File	Traverse Folder allows a user or group to access a folder nested within a tree, even if parent folders in that tree deny said user/group access to the contents of those folders. Execute File allows a user or group to run a program.
List Folder/ Read Data	List Folder allows a user or group to see objects (files and folders) inside a folder. Read Data allows a user or group to open and view a file
Read Attributes	Allows a user or group to view basic attributes of an object (read-only, system, archive, and hidden).
Read Extended Attributes	Allows a user or group to view the extended attributes of an object. For example: the summary, author, title, and so on for a Word document. These attributes vary from program to program.
Create Files/ Write Data	Create Files allows a user or group to create new objects within a folder. Write Data allows a user or group to overwrite an existing file.
Create Folders/ Append Data	Create Folders allows a user or group to nest folders. Append Data allows a user or group to add data to an existing file, but not delete data within that file or delete the file itself.

Special permissions

NTFS Access Limitations

Microsoft provides the following table to offer a more detailed understanding of what each permission can allow you to do. You should always refer to this table when assigning permissions. Try to assign the most restrictive possible permissions for each use case. A common bad practice in many IT business environments is to assign “full control” every time a user or group requests access to a file or folder.

Special Permissions	*Full Control*	*Modify*	*Read and Execute*	*List Folder Contents*	*Read*	*Write*
Traverse Folder/ Execute File	x	x	x	x		
List Folder/ Read Data	x	x	x	x	x	
Read Attributes	x	x	x	x	x	
Read Extended Attributes	x	x	x	x	x	
Create Files/ Write Data	x	x				x
Create Folders/ Append Data	x	x				x
Write Attributes	x	x				x
Write Extended Attributes	x	x				x
Delete Subfolders and Files	x					
Delete	x	x				
Read Permissions	x	x	x	x	x	x
Change Permissions	x					
Take Ownership	x					
Synchronize	x	x	x	x	x	x

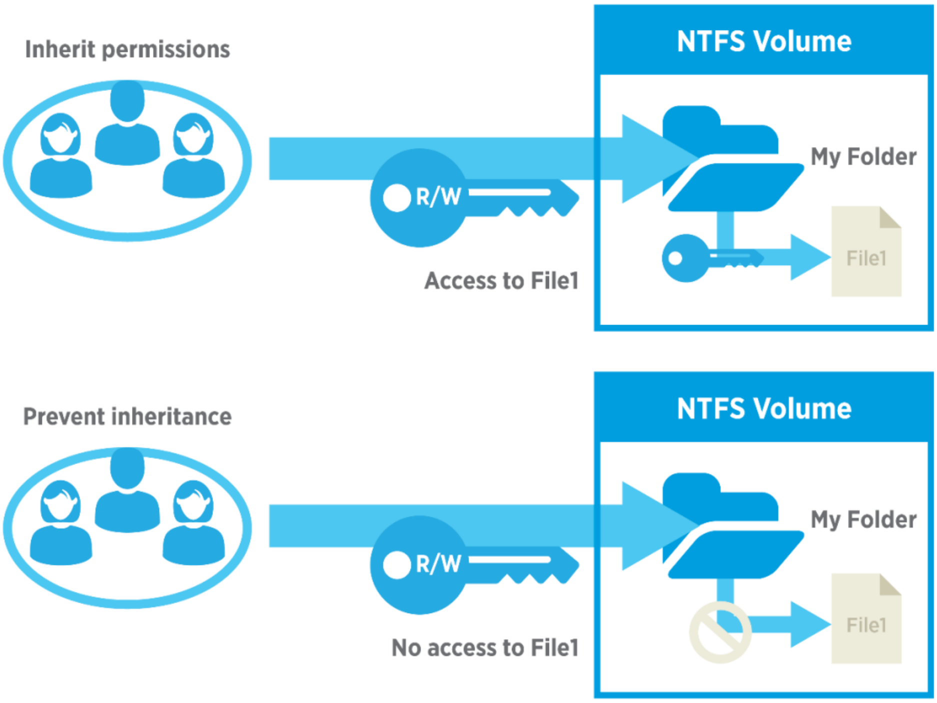
NTFS access limitations

Permission Inheritance

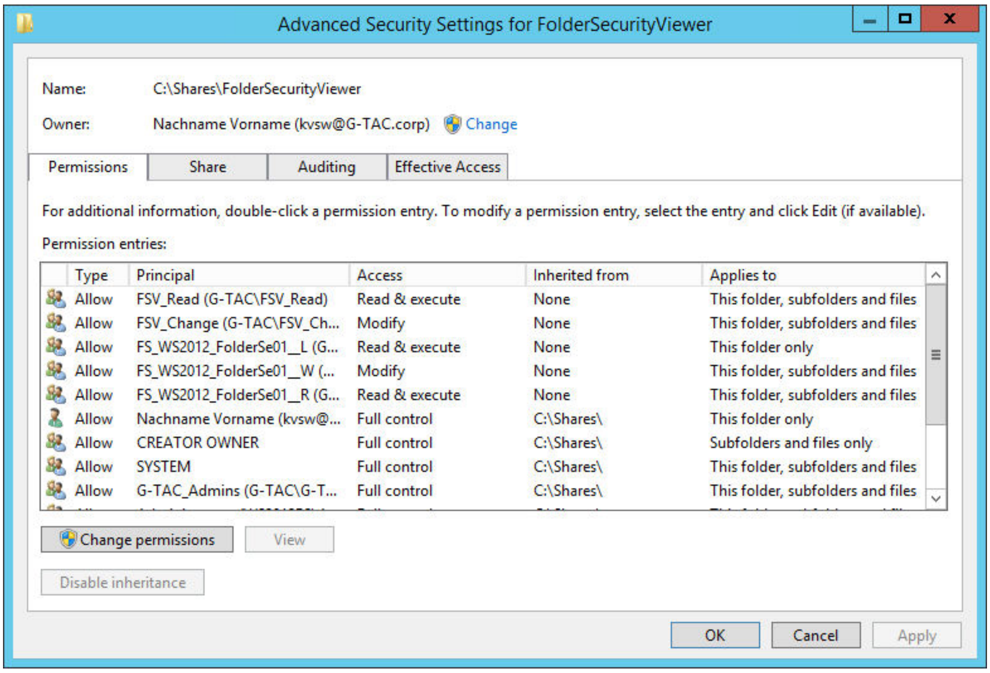
There are two types of permissions in Windows NT environments:

- Explicit: Permissions that are applied by default to an object upon its creation or by user action.
- Inherited: Permissions that are propagated to a child object. Inherited permissions facilitate the management tasks related to permissions assignment and ensure consistency among all the objects within a folder.

You must take into account that, by default, all objects created within the same folder inherit permissions from its respective parent folder. For example, if you create a folder called MyFolder, all subfolders and files within MyFolder will inherit its permissions automatically. In this order of ideas, MyFolder has explicit permissions and all subfolders and files in MyFolder have inherited permissions.



You can disable inheritance for any given file or folder by going to the security tab of its properties (as explained above) and clicking on **Advanced** and **Disable Inheritance.**



Advanced Security Settings for FolderSecurityViewer

Name: C:\Shares\FolderSecurityViewer

Owner: Nachname Vorname (kvsw@G-TAC.corp) Change

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Allow	FSV_Read (G-TAC\FSV_Read)	Read & execute	None	This folder, subfolders and files
Allow	FSV_Change (G-TAC\FSV_Ch...	Modify	None	This folder, subfolders and files
Allow	FS_WS2012_FolderSe01__L (G...	Read & execute	None	This folder only
Allow	FS_WS2012_FolderSe01__W (...	Modify	None	This folder, subfolders and files
Allow	FS_WS2012_FolderSe01__R (G...	Read & execute	None	This folder, subfolders and files
Allow	Nachname Vorname (kvsw@...	Full control	C:\Shares\	This folder only
Allow	CREATOR OWNER	Full control	C:\Shares\	Subfolders and files only
Allow	SYSTEM	Full control	C:\Shares\	This folder, subfolders and files
Allow	G-TAC_Admins (G-TAC\G-T...	Full control	C:\Shares\	This folder, subfolders and files

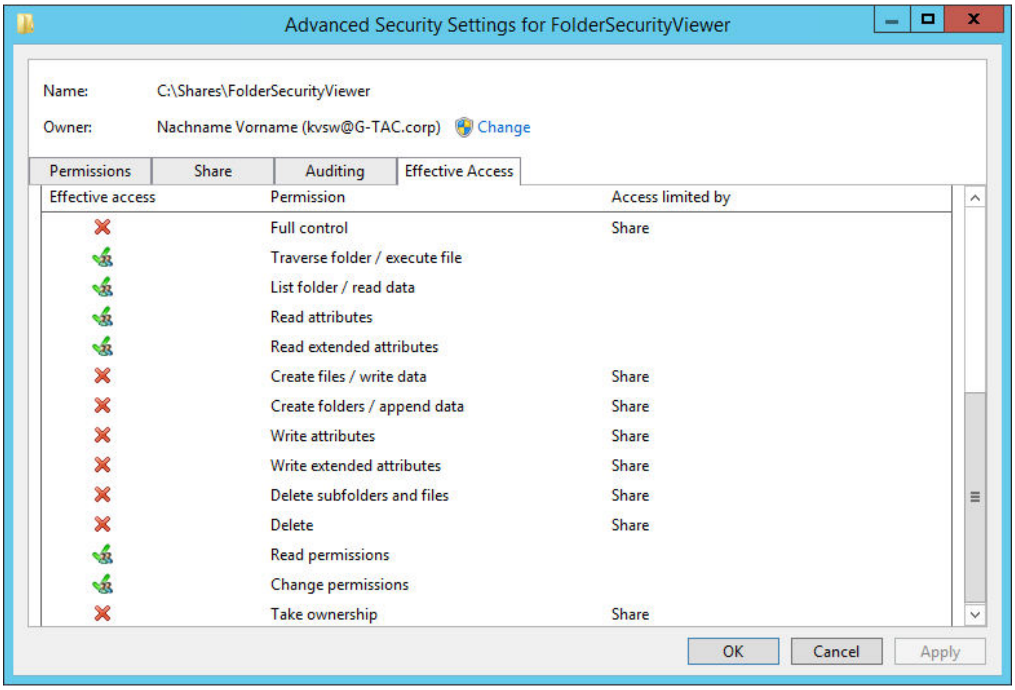
Disabling inheritance and replacing child object permissions in the Advanced Security Settings tab

When administrators and users start changing permissions and making regular changes, some files or folders can become inaccessible and users/groups that should have access to an object can lose their access. That’s why you can go back, at any time, to the default inherited state of any prior time by choosing the option “replace all child object permission entries with inheritable permission entries from this object”.

Effective Access

The effective permissions tab, found in the Advanced Security Settings Editor in earlier versions of Windows, was replaced with a tab called effective access in Windows Server 2012 R2 and Windows Server 2016, which lets you choose not only the user or group accessing the file or folder, but also the device accessing that file or folder.

This tab provides an overview of all the permissions assigned to a user or group in regards to accessing a certain object. For example, if John has “read” permissions for MyFolder and belongs to a group with “write” permissions, the effective access tab will show you that John has both “read” and “write” permissions for MyFolder.



Name: C:\Shares\FolderSecurityViewer

Owner: Nachname Vorname (kvsw@G-TAC.corp)

Effective access	Permission	Access limited by
✗	Full control	Share
✓	Traverse folder / execute file	
✓	List folder / read data	
✓	Read attributes	
✓	Read extended attributes	
✗	Create files / write data	Share
✗	Create folders / append data	Share
✗	Write attributes	Share
✗	Write extended attributes	Share
✗	Delete subfolders and files	Share
✗	Delete	Share
✓	Read permissions	
✓	Change permissions	
✗	Take ownership	Share

Effective access in Windows Server 2012 R2 and Windows Server 2016

Dynamic Access Control (DAC)

Though the interface has been improved, many of the underlying concepts of NTFS permissions have not changed over the years. The most notable changes are that the effective permissions tab has changed and dynamic access control (DAC) has been introduced.

DAC does not replace NTFS permissions, but does extend the capabilities offered by NTFS permissions and share permissions.

For example,

a user might have different permissions when they access a resource from their office computer than when they access that same resource using a laptop or over a virtual private network (VPN). In addition, access can be granted to a specific user only if said user's device meets the security requirements defined by administrators.