A Basic Introduction to NTFS Permissions

Author: Michael J. Leaver, 2BrightSparks

Introduction

In Windows, permissions are available on every file, folder, registry key, printer and Active Directory object. However, in this article we'll be concentrating on NTFS file and folder permissions. These permissions are available on NTFS file systems but not on FAT based file systems.

Permissions define what a user can and cannot do with a file or folder. For example they may be used to allow some users to read a file and disallow others from reading it. They could also be used to stop some users deleting or modifying files etc.

Basic Permissions

There are **basic** and **advanced** permissions.

The basic permissions map to one or more advanced permissions. For example, if you set the basic **Read** permission on a file then it means you have the following advanced permissions: List Folder/Read Data, Read Attributes, Read Extended Attributes and Read Permissions. Basic permissions provide a simpler and less granular way to set permissions. Another way to think of it is that basic permissions are groups of advanced permissions.

The basic permissions are:

Full Control: Users can read, modify, add, move, and delete files, as well as their associated properties and directories. In addition, users can change permissions settings for all files and subdirectories.

Modify: Users can view and modify files and file properties, including deleting and adding files to a directory or file properties to a file.

Read & Execute: Users can run executable files, including scripts.

Read: Users can view files and file properties.

Write: Users can write to a file.

Advanced Permissions

The advanced permissions are:

Traverse Folder/Execute File: For folders: Traverse Folder allows or denies moving through folders to reach other files or folders, even if the user has no permissions for the traversed folders. (Applies to folders only.) Traverse folder takes effect only when the group or user is not granted the Bypass traverse checking user right in the Group Policy snap-in. (By default, the Everyone group is given the Bypass traverse checking user right.) For files: Execute File allows or denies running program files. (Applies to files only). Setting the Traverse Folder permission on a folder does not automatically set the Execute File permission on all files within that folder.

List Folder/Read Data: List Folder allows or denies viewing file names and subfolder names within the folder. List Folder only affects the contents of that folder and does not affect whether the folder you are setting the permission on will be listed. (Applies to folders only.) Read Data allows or denies viewing data in files. (Applies to files only.)

Read Attributes: Allows or denies viewing the attributes of a file or folder, such as read-only and hidden. Attributes are defined by NTFS.

Read Extended Attributes: Allows or denies viewing the extended attributes of a file or folder. Extended attributes are defined by programs and may vary by program.

Create Files/Write Data: Create Files allows or denies creating files within the folder. (Applies to folders only). Write Data allows or denies making changes to the file and overwriting existing content. (Applies to files only.)

Create Folders/Append Data: Create Folders allows or denies creating folders within the folder. (Applies to folders only.) Append Data allows or denies making changes to the end of the file but not changing, deleting, or overwriting existing data. (Applies to files only.)

Write Attributes: Allows or denies changing the attributes of a file or folder, such as read-only or hidden. Attributes are defined by NTFS. The Write Attributes permission does not imply creating or deleting files or folders, it only includes the permission to make changes to the attributes of a file or folder.

Write Extended Attributes: Allows or denies changing the extended attributes of a file or folder. Extended attributes are defined by programs and may vary by program. The Write Extended Attributes permission does not imply creating or deleting files or folders, it only includes the permission to make changes to the extended attributes of a file or folder.

Delete Subfolders and Files: Allows or denies deleting subfolders and files, even if the Delete permission has not been granted on the subfolder or file. (Applies to folders.)

Delete: Allows or denies deleting the file or folder. If you do not have Delete permission on a file or folder, you can still delete it if you have been granted Delete Subfolders and Files on the parent folder.

Read Permissions: Allows or denies reading permissions of the file or folder, such as Full Control, Read, and Write.

Change Permissions: Allows or denies changing permissions of the file or folder, such as Full Control, Read, and Write.

Take Ownership: Allows or denies taking ownership of the file or folder. The owner of a file or folder can always change permissions on it, regardless of any existing permissions that protect the file or folder.

Assigning, Allowing and Denying Permissions

Permissions are assigned **explicitly** or by **inheritance**. For example, a file could inherit its permissions from its parent folder. This makes managing permissions simpler as you only need to change one folder's permission instead of all the files in a folder. You can also set explicit permissions for a file or a folder. For example, a file could still inherit its permissions from its parent folder but you may also want to give extra permissions to a specific user.

You can **allow** or **deny** a permission. Deny beats Allow if they are applied on the same file or folder. If the permissions are inherited, then the Allow and Deny work a bit differently. It is based on a hierarchy:

1. Explicit Deny
2. Explicit Allow
3. Inherited Deny
4. Inherited Allow

These are checked, by Windows, from first to last, and once one is matched then that security is used. For example, if the inherited permissions on a file are that you are denied read permission, but you are explicitly given read permission on the file, then that explicit permission overrides the inherited deny permission.

The security applies to both **users and groups**. Users can be members of one or more groups. For example, you could have a group for accountants, editors, programmers, etc. You can then base your permissions on groups instead of specific users. This makes managing the permissions much simpler as a new employee can simply be added to the appropriate groups. They can then access files and folders based on their group and no permissions need to be changed.

Files and folders have **ownership**. When a file or folder is created Windows gives Full Control to the owner (the creator of the file or folder). You can change ownership, but the user or group changing the ownership needs the **Full Control** or **Take Ownership** permission.