Computer Systems

# UD 07. WINDOWS ADMINISTRATION -PART 3-

**COMPUTER SYSTEMS**
**CFGS DAW**

Sergio García / Alfredo Oltra
sergio.garcia@ceedcv.es
alfredo.oltra@ceedcv.es

2022/2023

Versión:220602.1643

## Licencia

## Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

Importante

Atención

Interesante

## INDEX

# 1. INTRODUCTION

In this part we are going to study the basis of NTFS permissions for Windows Systems. This is very important and useful to secure our Windows systems. The knowledge about NTFS permissions is valid for any Windows System with NTFS file system.

# 2. USERS AND GROUPS IN WINDOWS 10

In Windows 10 you can reference to individual users or to groups. A group is an element that contains a list of users and/or other groups.

For example, suppose we have pupils for class A and class B:

- If we have users alumno1classA, alumno2classA, alumno3classA, we can group them in a group called “CEEDClassA”.
- If we have users alumno1classB, alumno2classB, alumno3classB, we can group them in a group called “CEEDClassB”.
- If we want a group of all pupils of both groups, we can do it by two ways:
  - Create a group “CEEDClasses” and add manually all users. If a new user is added to a class, you have to add it in its class group and in “CEEDClasses”.
  - Create a group “CEEDClasses” and add only groups “CEEDClassA” and “CEEDClassB”. If you add a new user, you only have to add it to its class group.

In this video, you have a video tutorial of how to create and manage groups in Windows 10 https://www.youtube.com/watch?v=SKpzdSU6DOA

# 3. NTFS PERMISSIONS

## 3.1 Introduction

In modern Windows systems with NTFS file system, you can set permissions to files and folders. There are two kinds of permissions:

- Individual NTFS permissions: there are a lot of permissions for each possible action in files and folders. Using those permissions let you to set a high detail on what you let to do to users in your files and folders.
- Special NTFS permissions: actually Special NTFS permissions are “groups of individual permissions”. They exist to do management easier summarizing the most used group of individual permissions.

## 3.2 Ownership and administrators “take ownership”

In NTFS file system each file or folder has an owner.

Initially, the owner of a file or a folder is its creator, but it can be changed.

The owner has one special right about their own files and folders: **he always can change permissions**. It is the only special right. For other matters, it is a normal user.

> When you create a file or a folder, by default Windows set “Full control” permission to creator.

For NTFS permissions, administrators of the system are normal users too. They only have a special right in the system: **they can “take ownership” of any file or folder**, but when we try to access to a file or folder they are normal users.

> Why owners and administrators are limited?
>
> This is in order to avoid fatal mistakes.
>
> For example, if an administrator make a mistake and try to modify a file what is supposed can’t be modified by its permission configuration, they couldn’t be able to modify it by accident.
>
> An administrator can become owner of any file. Virtually administrator has full power (an administrator can take ownership and modify permissions of any file or folder).

In this video you can watch a sample of taking ownership https://www.youtube.com/watch?v=YWgDDip5Bqo

## 3.3 Setting permissions in Windows 10

In Windows 10 you can set permissions to users and groups for a resource. Each resource has a list with their permissions. That list includes a list of groups and users that permissions apply to and it is called **ACL** (Access Control List).

A permission for a group or users could take 3 states:

- **Granted**: the permission is granted. It is necessary (but not sufficient, view point “Grant permissions algorithm”) to get the desired permission.
- **Denied**: a permission is denied. If you have a permission denied, you could never get that permission.
- **Not set:** the permission is not set (not granted or denied)

In this video you can watch how to set permissions in Windows 10 https://www.youtube.com/watch?v=3LnnvbpO9NI

## 3.4 Inherit permissions

Windows has a mechanism of inheritance for permissions.

If you have a resource (file or folder) inside a folder, by default you inherit their permissions. It means that if you have set permissions for folder “A” and “B” is a folder inside “A” with inherit activated, “B” has the same permissions that “A”.

If you set manually permissions that collide with inheritance permissions, the “manual permissions” go first than “inherit permissions” (View “Grant permissions algorithm”).

By default, inheritance is activated. Inheritance could be deactivated individually for each resource.

More info about inheritance:

https://technet.microsoft.com/en-us/library/cc726071(v=ws.11).aspx

## 3.5 Grant permissions algorithm

To grant or deny a permission, Windows uses the following algorithm:

- He obtains the ACL entries where your user match (its own entry if it exists and group entries where your user is member of).
- To grant the permission, you need to be granted in at least one entry and not denied in any entry.
- If you don’t get grant or denied in any entry, permission is denied.

For example, user “pepito” that is member of group “Friends”.

- If permission is granted to “pepito” and it is granted to “Friends”, permission is granted.
- If permission is granted to “pepito” and it is not set to “Friends”, permission is granted.
- If permission is granted to “Friends” and that permission is not set for “pepito”, permission is granted.
- If permission is granted for “Friends” but it is denied for “pepito”, permission is denied.
- If permission is granted for “pepito” but it is denied for “Friends”, permission is denied.
- If permission is not set for “pepito” and is not set for “Friends”, permission is denied.

### 3.5.1 Gran permissions algorithm with inheritance

If there are inherit permissions, algorithm is applied first to “manually set permissions”.

Only in the case that the permissions is not set for any entry in “manually set permissions”, algorithm is applied to “inherit permissions”.

For example, folder A with folder B inside and inheritance activated in folder B. User “pepito” want to obtain a permission for folder B.

- If permission is granted in A and is not set in B, permission is granted.
- If permission is granted in B and denied in A, permission is granted.
- If permission is granted in A and denied in B, permission denied.
- If permission is not set in A and is not set in B, permission denied.

# 4. PERMISSION LIST

## 4.1 Individual permissions

The list of individual permissions is:

- **Traverse Folder/ Execute File:** in a file, you can execute it. In a folder, you can go “inside” the folder.
- **List Folder/ Read Data**: in a file, you can read it. In a folder, you can list files and folders inside.
- **Read Attributes**: you can read attributes in a file or a folder.
- **Read Extended Attributes**: you can read extended attributes in a file or a folder.
- **Create Files/ Write Data**: in a file, you can write it (destroying old content), in a folder you can create new files.
- **Create Folders/ Append Data**: in a file, you can write it (appending new content), in a folder you can create sub folders.
- **Write Attributes**: you can write attributes in a file or a folder.
- **Write Extended Attributes**: you can write extended attributes in a file or a folder.
- **Delete Subfolders and Files:** in a folder, you can delete files and subfolders, even if you don’t have deleted permission for each element.
- **Delete:** with this permission, you can delete the file or folder affected.
- **Read Permissions:** you can read permissions in a file or a folder.
- **Change Permissions:** you can change permissions in a file or a folder.
- **Take Ownership:** you can take ownership of a file or a folder.

- **Synchronize:** used by multi-thread programs, it is related with synchronization between threads.

Individual permissions and their equivalence to special permissions can be obtained in detail in

https://technet.microsoft.com/en-us/library/cc783530(v=ws.10).aspx#w2k3tr_randp_how_tfqi

### 4.2 Special permissions

In this point we show a table that summarize what individual permissions are included in special permissions.

| Special Permissions | Full Control | Modify | Read & Execute | List Folder Contents (folders only) | 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 |

## 5. ADDITIONAL MATERIAL

[1] Windows Training
https://www.microsoft.com/en-us/learning/windows-training.aspx

[2] NTFS Permissions
http://examples.oreilly.com/0636920032748/NTFS_Permissions.pdf

## 6. BIBLIOGRAPHY

[1] Windows 10

https://en.wikipedia.org/wiki/Windows_10

[2] Microsoft support

https://support.microsoft.com/

[3] Permission List

https://technet.microsoft.com/en-us/library/cc783530(v=ws.10).aspx#w2k3tr_randp_how_tfqi

[4] Inherit permissions

https://technet.microsoft.com/en-us/library/cc726071(v=ws.11).aspx