

## Lab 7 – Network File Shares and Permissions

In this lab we will learn about dynamic disks, file server configuration and NTFS security permissions. Specifically, we will configure a second hard drive on our server to work as a dynamic disk and install file server features in Windows Server 2016. We will also create shared folders and assign them to users and groups with corresponding NTFS and shared access permissions.

### Discussion

#### *Basic vs. Dynamic Disks*

In Windows Server 2016, there is support for two disk configuration types—**basic disk** and **dynamic disk**. Understanding of these configurations will help you to effectively configure and troubleshoot disk storage.

Basic Disk - The basic disk, is similar to the disk configuration used in older versions of Windows OS 9X and NT. It is a physical disk with primary and extended partitions. As long as you use an appropriate format, client Windows systems all the way back to DOS can access basic disks.

Always use **basic** disks rather than dynamic on computers running **older** versions of MS OSes such as MS-DOS, Windows 95, 98, ME, NT 4.0, or XP that are configured to dual-boot with the Windows Server family of operating systems. These older operating systems cannot access the data stored on dynamic volumes.

Dynamic Disk: A Dynamic Disk is a physical disk with neither partitions nor logical drives. Instead, it contains only dynamic volumes that you create in the Disk Management console. Regardless of what format you use for the file system, only Win2K and later operating systems can access dynamic volumes directly. Computers running older systems **can** access dynamic disks **remotely** when connected to the shared folders over the network.

One limitation of basic disks is the four-partition limit (three primary partitions, and one extended partition with logical drives). Dynamic disk partitions (volumes) are no longer restricted to four. Other advantages to using dynamic disks include data protection techniques such as striping or mirroring volumes. Dynamic disks also allow administrators to extend the volumes and make changes to the disk without rebooting the computer.

## NTFS Permissions

NTFS file permissions are used to control the access to files on a computer. This includes everything from reading a file to modifying and executing the file. There are five NTFS file permissions as listed in the table below:

TABLE 1: NTFS FILE PERMISSIONS	
NTFS Permission	Allowed Access
	Allows the user or group to ...
Read	... read the file and view its attributes, ownership, and permissions set.
Write	... overwrite the file, change its attributes, view its ownership, and view the permissions set.
Read & Execute	... run (execute) the application. In addition, the user can perform all duties allowed by the Read permission.
Modify	... modify and delete a file including perform all of the actions permitted by the Read, Write, and Read and Execute NTFS file permissions.
Full Control	... change the permission set on a file, take ownership of the file, and perform actions permitted by all of the other NTFS file permissions.

Some permissions build on others. For example, Modify permission implies that a user automatically gets to Read, Write, and Read & Execute. So if a user needs all access, except taking ownership and changing permissions, to a file the Modify permission can be granted. This saves us from having to assign multiple permissions to a file or group of files.

NTFS folder permissions determine what access is granted to a folder and the files and subfolders in that folder. These permissions can be assigned to a user or group. The following table defines each NTFS folder permission and its effect on a folder.

TABLE 2: NTFS FOLDER PERMISSIONS	
NTFS Permission	Allowed Access
	Allows the user or group to ...
Read	... view the files, folders, and subfolders of the parent folder. It also allows the viewing of folder ownership, permissions, and attributes of that folder.
Write	... create new files and folders within the parent folder as well as view folder ownership and permissions and change the folder attributes.
List Folder Contents	... view the files and subfolders contained within the folder.
Read & Execute	... navigate through all files and subfolders including perform all actions allowed by the Read and List Folder Contents permissions.
Modify	... delete the folder and perform all activities included in the Write and Read & Execute NTFS folder permissions.
Full Control	... change permissions on the folder, take ownership of it, and perform all activities included in all other permissions.

Notice that the major difference between the file and folder permissions is the “List Folder Contents” permissions for NTFS folders. The NTFS folder permission allows us to limit a user's ability to browse through a tree of folders and files. This is useful when trying to secure a specific directory such as an

application directory. A user must know the name and location of a file to read or execute it when this permission is applied to the parent folder of the file.

Applying Multiple NTFS Permissions - Multiple permissions can be assigned to a single user account. They can be assigned to the user account directly, or to a group that the user is a member of. When multiple permissions are assigned to a user account, unexpected things can happen. To prevent headaches, we are going to discuss the rules for assigning multiple NTFS permissions to a single user or group. This will include how file and folder permissions work together, and how denying a specific permission can affect a user's allowed access.

NTFS permissions are **cumulative**. This means that a user's effective permissions are the result of combining the user's assigned permissions and the permissions assigned to any groups that the user is a member of. For instance, if a user is assigned Read access to a specific file, and a group that the user account is a member of has Write permissions, the user will be allowed Read and Write NTFS permission for that file.

NTFS file permissions override or take priority over NTFS folder permissions. A user account having access to a file can access that file even if that user does not have access to the parent folder of that file. However, a user would not be able to do so via the folder, because that requires the "List Folders Contents" permission. When the user attempts to access the file, he or she must supply the full path to it. The full path can either be the logical file path (F:\MyFolder\MyFile.txt) or use the Universal Naming Convention (UNC). To access the file via UNC the user must supply the server name, share, directory, and file, for example:

**\\MYSERVER\Win2kShare\MyFolder\MyFile.txt**

If the user has access to the file but does not have an NTFS folder permission to browse for that file, the file will be invisible to the user and he or she must supply the full path to access it.

"Deny" Overrides All Other Permissions: If we assign a Deny value to an NTFS "permission" for a file to user, any other setting that gives the same user permission to access the same file will be negated. Use caution when assigning "deny" to control access to a resource. For instance, if a user has access to a file or folder as being a member of a group, denying permission to that user stops all other permissions that the user might have to the file or folder.

This can be hard to troubleshoot on a large network with thousands of users and groups. This concept of permission denial has not changed through the evolution of the Microsoft Windows operating systems and NTFS. The figure shows an example of multiple NTFS file and folder permissions: and what happens to the user's effective permissions.

User A is a member of both Group 1 and Group 2. Group 1 settings allows access to Folder A and both of the files within that folder. Group 2, on the other hand, denies access to a specific file, File 1. When a user account is assigned a "deny" access to a file or folder, all other permissions granting that user access to that file or folder are overridden: user A's combined access to File 1 is a "No access".

Figure 1  
Deny permission overrides all other permissions

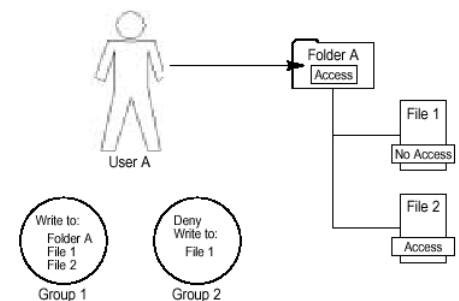
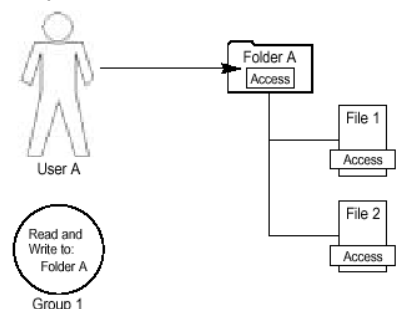


Figure 2  
NTFS permission inheritance



## Understanding Inherited NTFS permission

By default, when NTFS permissions are assigned to a **parent folder**, all of the same permissions are applied or propagated to the **subfolders** and files of that parent folder. An example of NTFS permission inheritance is shown in the following figure:

By getting “Read and Write” access to Folder A, user A gets the same access to File 1 and File 2 in Folder A. i.e. files and subfolders inherit NTFS permissions from their parent folder.

As the Windows Server 2016 administrator, you assign NTFS permissions to a folder. All current subfolders and files within that folder inherit those same permissions. In addition, any new files or subfolders created within that parent folder assume the same NTFS permission of that parent folder.

It is possible to stop this propagation of NTFS permissions. We can make files and subfolders in a parent folder unable to inherit the NTFS permissions of their parent folder. Keep in mind that the directory or folder level in which you decide to prevent the default NTFS permission inheritance becomes the **new parent folder** for NTFS permission **inheritance**.

## Planning NTFS Permissions

It is important to plan out our Windows Server 2016 network: the first step, the Active Directory and Windows Server 2016 domain infrastructure we saw in previous labs is very important, but a plan for NTFS permissions should also be well thought out before we implement our network.

Having a plan for file permissions in a Windows network saves time and effort in any organization. A set of well-planned NTFS permissions makes it easier to manage access to shared files. Use the following guidelines to plan NTFS permissions on your Windows network. Some steps are not directly related to NTFS permissions themselves, but they help organize storage on our network and make sure those resources are secure.

Seven guidelines to plan NTFS permissions:

1. Organize data on a Windows network in manageable units. Separate users' home directories from applications and public data. Try to keep data in centralized units. For instance, group all of the home directories into one folder and place them on an NTFS volume away from other data. This benefits us as we would not have to assign NTFS permissions to files, but only to the grouped folders. In addition, backup strategies become less complex. When application files are grouped separately they do not have to be backed up with the home directories. Organizing your data can make many things easier to manage, including assigning NTFS permissions.
2. Assign users the minimum level of access that is required. If a user needs only to read a file, grant only the Read permission to the resource that they require access to. This precludes the possibility of a user damaging a file, such as modifying an important document or even deleting it.
3. When a group of users require the same access to a resource, create a group for those users and make each a member of that new group. Assign the NTFS permissions required to that resource to the newly created group. Avoid assigning NTFS permissions to individual users when possible and only assign them to groups. This saves a lot of time.
4. When assigning permissions to folders with working data, use the Read & Execute NTFS folder permission. This should be assigned to a group containing the users that need to access this folder, and to the Administrators group. This will allow the users to work with the data but will also prevent them from deleting any important files in the folder.

5. When assigning NTFS permissions to a public data folder, use the following criteria as a guideline. Assign the Read & Execute and Write NTFS permissions to the group containing the users that need access to the public data folder. The Creator Owner of the folder should be assigned the Full Control NTFS permission. Any user on the network that creates a file, including one in a public data folder, is by default the Creator Owner of that file. After that file has been created, the Windows server administrator can grant NTFS permissions to other users for file ownership. If the Read & Execute and the Write NTFS permissions are assigned to group of users that need access to the public data folder, they have Full Control to all files that they create in the public data folder and can modify and execute files created by other users.
6. Do not explicitly deny NTFS permissions to a group or user. Assigning a “deny” is not recommended on Windows network, because the “deny” only affects that specific resource, and leave all others unaffected. If all new resources are managed this way, we will lose track of what is denied for what groups. Troubleshooting permissions problems will take longer.
7. User education is always a good idea. If users have a basic understanding of the NTFS permissions and how to secure resources on a network, they can assign and manage their own files. User education does take a bit of time but it does pay off in the end if done correctly.

When planning how to organize your data on a Windows network, consider the effects of NTFS permissions carefully. Every business or organization is different, but following the simple guidelines above will make managing your resources in a secure environment much easier.

### ***Working With NTFS Permissions***

When a new volume is created and formatted with the NTFS 5.0 file system in Windows Server 2016, by default, the Full Control NTFS permission is granted to the Everyone group. Of course, this should be changed as soon as possible. The reason is that allowing Everyone full control means just that, everyone is free to change everything! That includes guests, if the Guest account is enabled, and even anonymous Internet users, if security settings on the firewall are such that they can access files on that server. By default, even though you are running NTFS, no security at all is applied. The approved NTFS permission plan should be implemented immediately.

If your organization does not yet have an NTFS permission plan, start by changing the access for the Everyone group from Full Control to Read. Then begin assigning the appropriate NTFS permissions to users as needed.

# Activities

## *Installing the file server on Windows Server 2016*

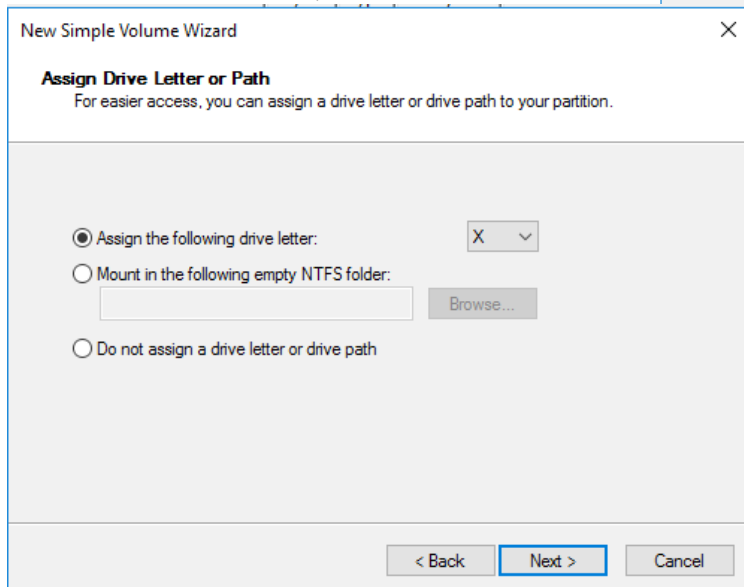
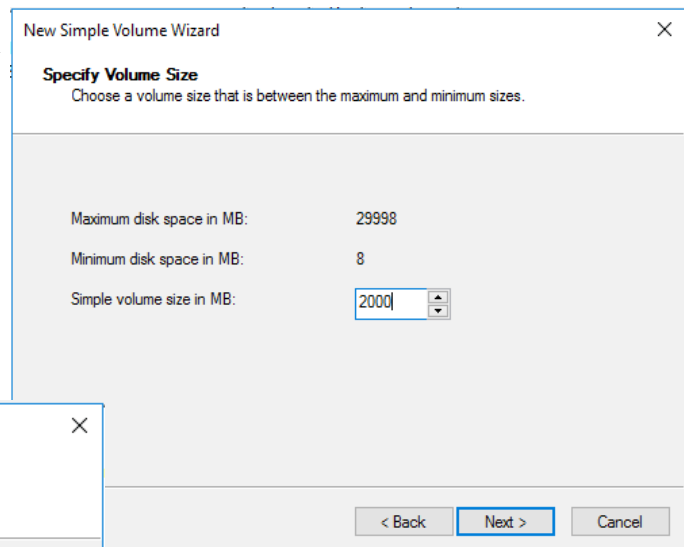
- Start Server Manager and select Manage → Add Roles Features → Next
- Make sure “Role-based or feature-based installation” is selected and click Next, and Next again.
- “Open” File and Storage Services and then open “File and iSCSI Services” and select File Server Resource Manager (if asked to add more roles, click Add Features) and click Next.
- Select Windows Search Service and click Next → Install.
- Your drive (most likely C) will be selected by default.

## *Formatting partitions, configuring it as a dynamic disk, creating dynamic volumes on the drive.*

- Start MMC and open your console. From **Computer Management, Storage**, select **Disk Management** and do the following steps on Disk 0:
- There should be at least 8GB of free space at the end of the disk.
- We will create one drive of size 2 Gig for each workstation team and one that will be shared by all teams. The rest of the disk space will be left for future expansion.
- Create a new partition (called **Volume**) on **Disk 0**. To do this, right click the Unallocated part in the right pane of **Disk 0** and select **New Simple Volume**. The **New Volume** wizard will appear. Click **Next**

On the following window, set the **Select the amount of space in MB** to 2 Gig (2000 Meg) as shown here and click **Next**.

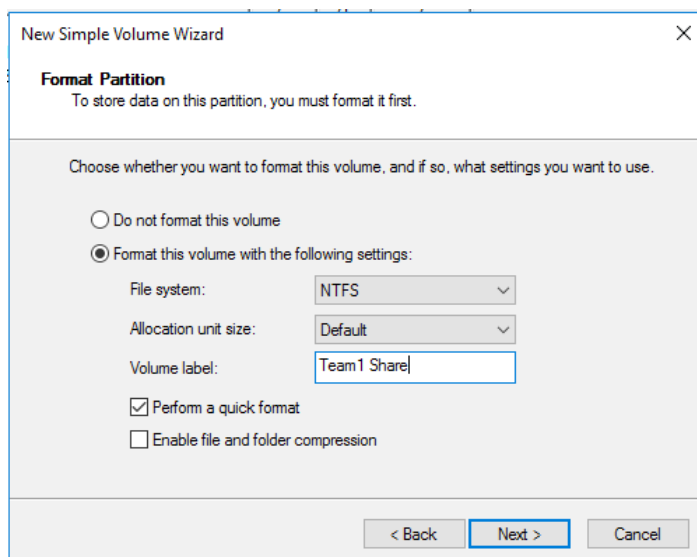
- Select a drive letter on the next screen. Use the upper drive letters **X**, **Y** and **Z** for the drives you create - start with **X** as the drive letter for the first partition and then **Y** for the second, etc.



Next format the partition using NTFS quick format as shown here. Set the Volume label to be the team's name + "Share" as shown here (use **X** for Team1, **Y** for Team2, and **Z** for All Teams share). Make sure "Perform a quick format" is checked.

Click **Next**, review the settings and if everything looks good click **Finish**.

Once you are done with all three drives, you should see the following:



**New Simple Volume Wizard**

**Format Partition**  
To store data on this partition, you must format it first.

Choose whether you want to format this volume, and if so, what settings you want to use.

☐ Do not format this volume

☒ Format this volume with the following settings:

File system: **NTFS**

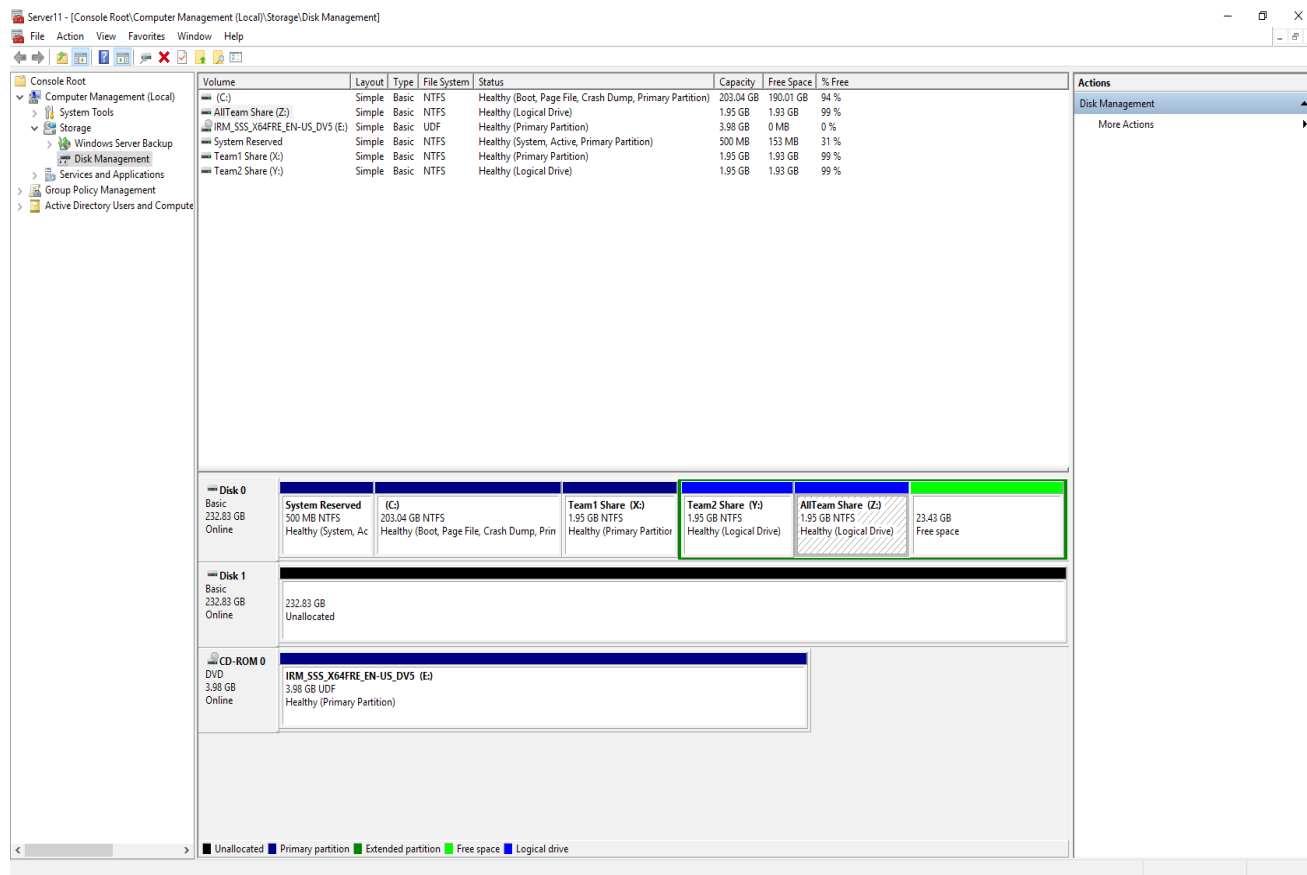
Allocation unit size: **Default**

Volume label: **Team1 Share**

☒ Perform a quick format

☐ Enable file and folder compression

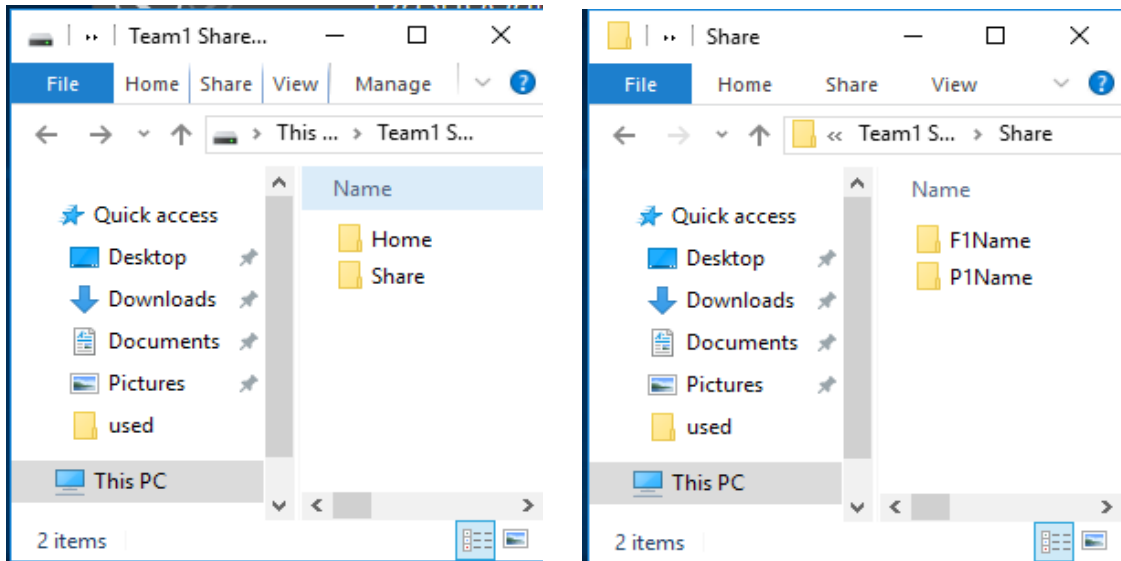
< Back   **Next >**   Cancel



**CONTACT ME AT THIS POINT AND DEMO YOUR SETUP. YOU WILL NEED A SIGN OFF ON THIS.**

## Creating shared areas

- We will use drive **X** for Team 1, drive **Y** for Team 2 and drive **Z** for the All teams share area.
- Open the drive **X** from File Explorer → This PC and create two folders named **home** and **share**. We will use the **home** folder for separate personal areas for each user and the **share** folder will be used by all users in Team1. Open the folder **home** and create one folder for each team member, Once you are done, you should have directory structure as shown below:

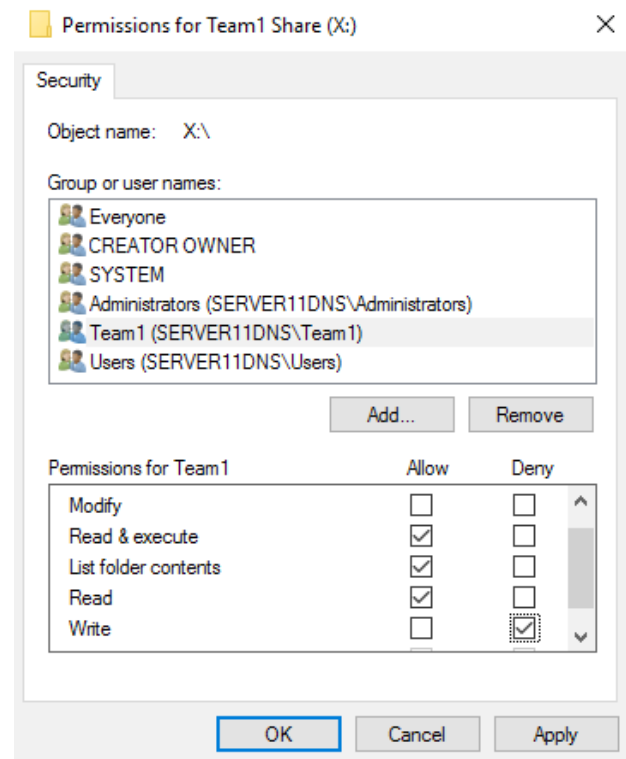
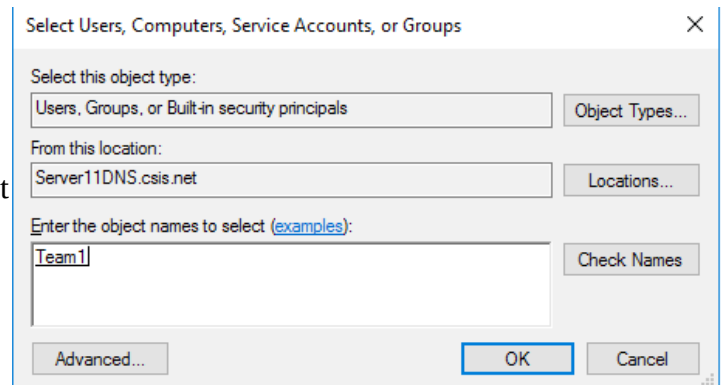


- Add a dummy text file in each directory so that we can test access permissions later. To create a text file, right click in the folder area and select **New → Text Document**. Use Notepad to add some dummy content to each file.
- Repeat the above steps for the **Y** drive – you can copy the folders in X, paste them into Y, and change the folder names in Y:\home.
- Create a dummy file in **Z**. You do not need to create any directories for **Z**. We will share the whole drive with all team members.

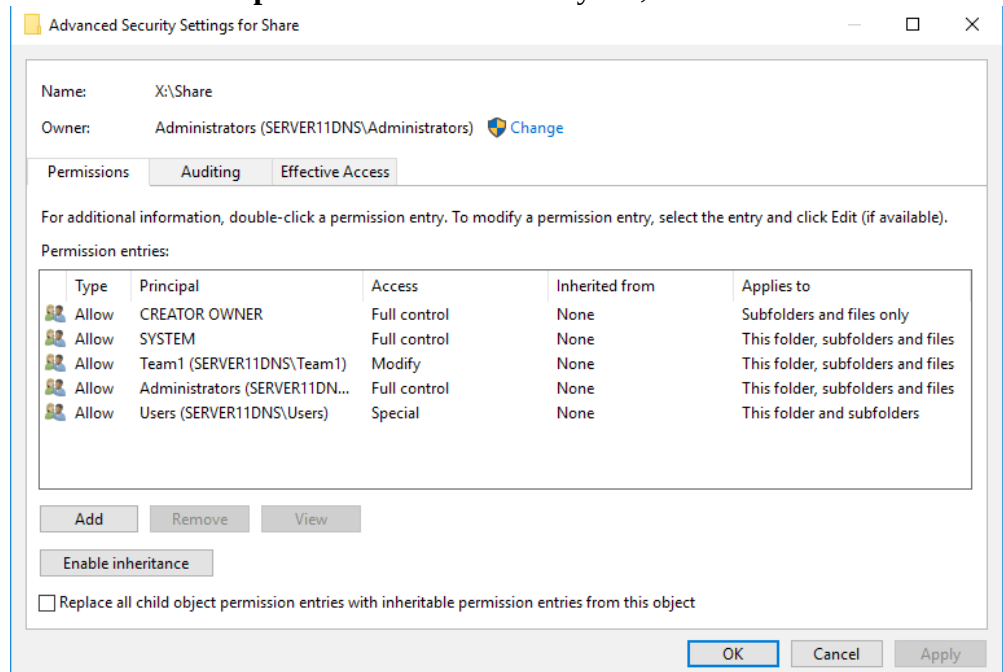


**Setting up NTFS permissions - (The following steps are for the X drive. Follow the same steps for Y)**

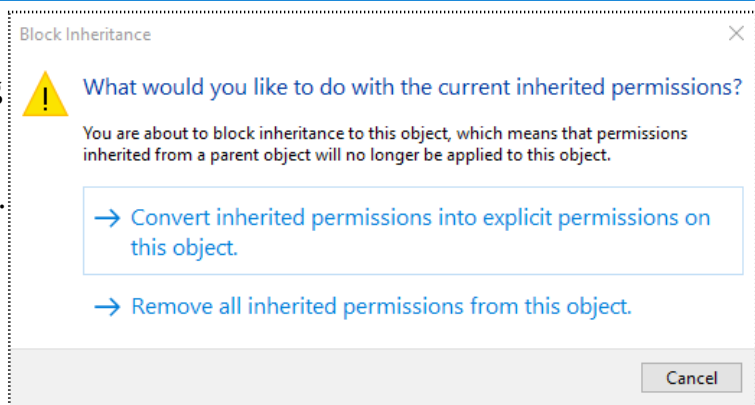
- Right click on the drive and select Properties. Select the Security tab and you should see the following default settings:
  - **Everyone: None**
  - **CREATOR OWNER: None**
  - **SYSTEM: Full Control, Modify, Read & Execute, List Folder Contents, Read, Write**
  - **Administrators: Full Control, Modify, Read & Execute, List Folder Contents, Read, Write**
  - **Users: Read & Execute, List Folder Contents, Read**
- Modify the security settings so that the only the **Users group** has **List Folder Contents** permissions. Make sure the **Everyone group** permissions are all unchecked - all allow boxes should be empty.
- Add the Team 1 group to the list: click **Edit → Add →** in the **Enter the object names to select** text area enter the first few characters of the team's name. Click **Check Name**, select the team from the list and click **OK**. Note: if you type in a complete unique user or team name, clicking Check Name will simply underline the name to show that the system recognizes it.
- Set the permissions for the team as shown here. This will allow all the team members to read the contents of the shared drive, but they will not have the ability to add to or change them. Click **OK**, read the warning and answer **Yes**.



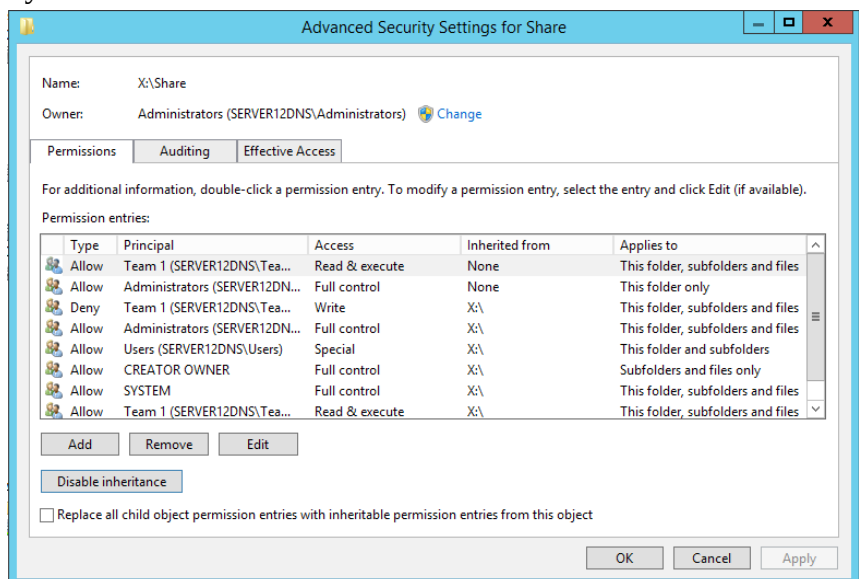
- Navigate to the **share** folder and select **Properties**. Click the Security tab, select Team 1 → **Advanced**. You should see a permissions tab similar to the following:



- Select **Team 1** → select **Disable inheritance** → a new “Block Inheritance” window will pop up asking you what to do. Choose **Convert inherited permissions into explicit permissions on this object** → OK → Yes.

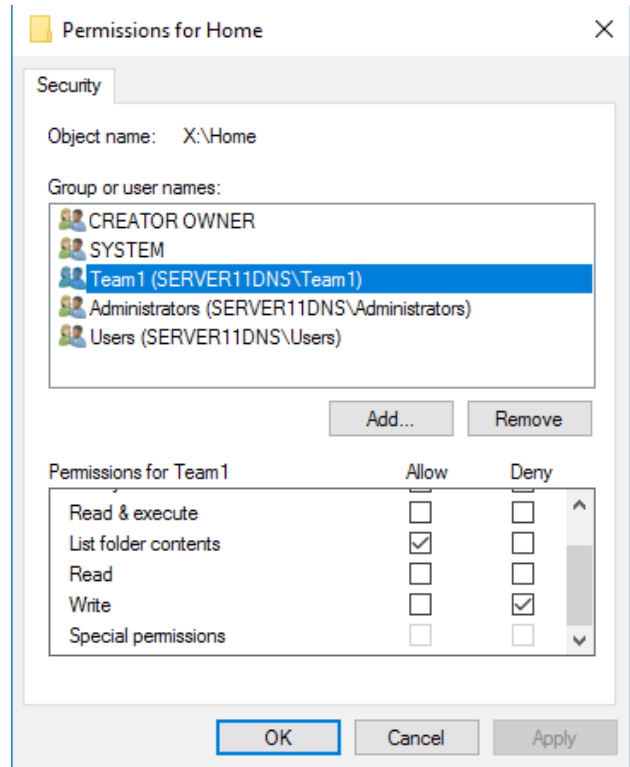


- Select Team 1 → Edit → select Modify and Write permissions for the team → OK.



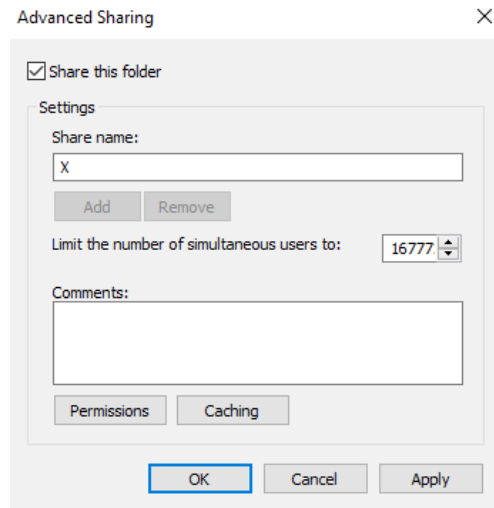
Disable Inheritance for **home**.

- Since the **home** folder is not a shared area, users may not want other team members to read their stuff. To remove these permissions, right click on the **home** folder, click Properties → Security → Edit, and for Team 1 remove the **Read and Execute** and **Read** Permissions
- Enter **the home** folder, right click on each user's folder → Security tab and do the following:
  - **Add** the corresponding user to the list of **Group and users names** and give them **Modify** and **Write** permissions.
- Now, do the above set-up steps for drive Y. Remember to set this drive up for use by Team 2.
- Setup the permissions for the **Z** drive in a similar manner. Remember, that the **Z** drive should be setup such that the **All Teams group** will have **Modify**, **Read & Execute**, **List Folder Contents**, **Read** and **Write** permissions.



**Sharing the drives - (The following are the steps for X. Follow the same steps for Y and Z)**

- Right click on the **X** drive and select **Properties** → **Sharing** → **Advanced Sharing**.
- Check **Share this folder**, then name it **X**

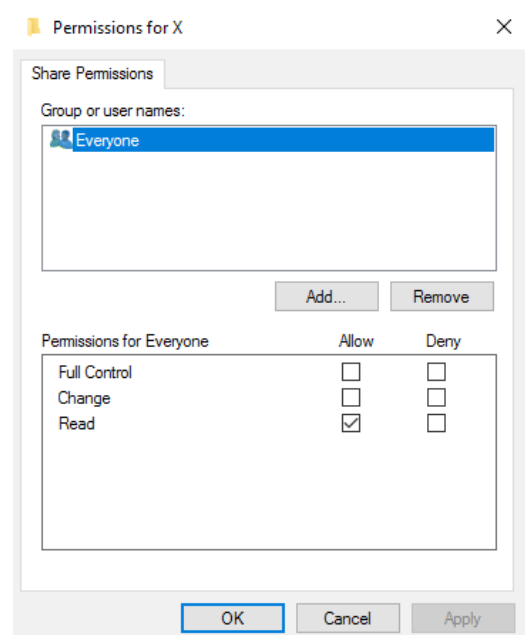


Next, we have to set the **Share** Permissions. These are different from the **NTFS** permissions we set earlier. These only apply for network access to the drive.

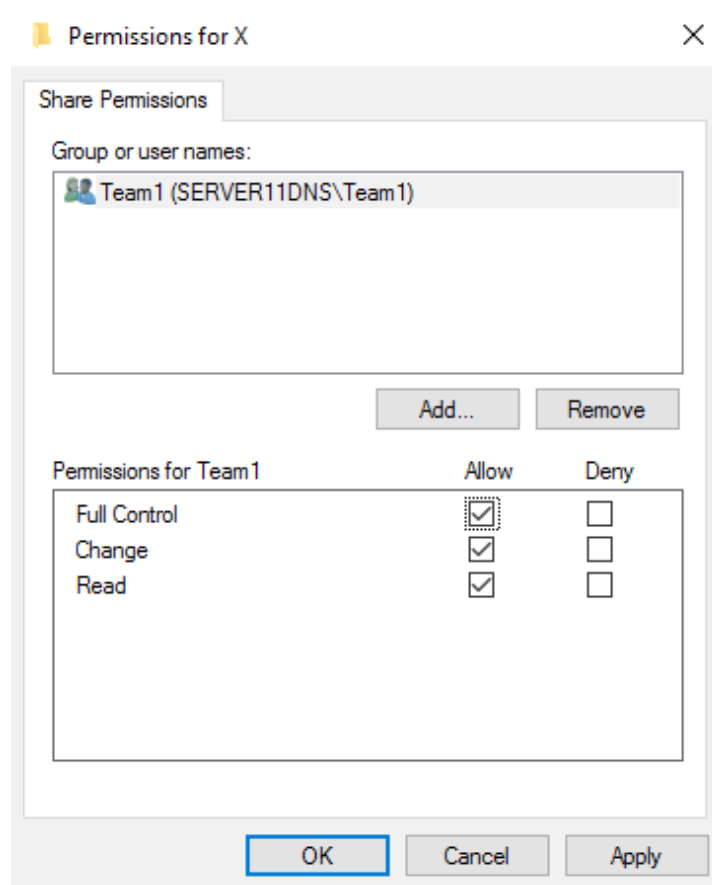
(The **NTFS** permissions are for local access)

When accessing a folder through a network the network permissions will be the more restrictive of the **shared** and **NTFS** permissions. For this reason, we will generally make the **shared** permissions a bit more relaxed and force stricter permissions at the **NTFS** level since **NTFS** offers a finer set of permissions and also applies to local access.

Click the **Permissions...** button and you should see the following:



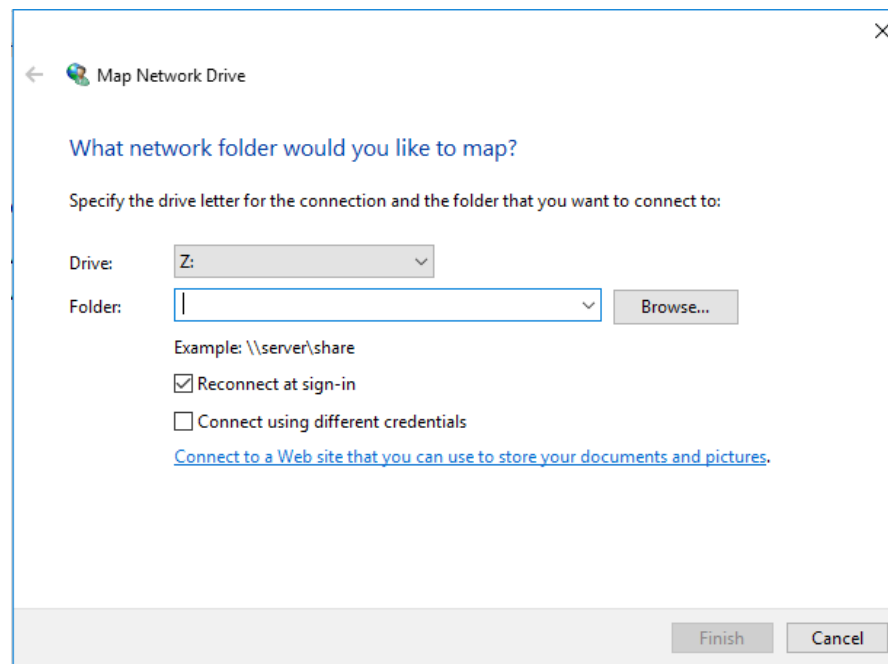
- Remove **Everyone group** from the list and add Team1 to the list and give them full control permissions. Once you are done you should see something similar to the following



- You should now check and see if you have done all of this correctly.
- From each of the workstations, login to the system as one of the partners and try opening and modifying the drives and the corresponding folders and files by clicking on start and then network. Write down what you were able to do and what you could not do. Explain your results in your report.
  - NOTE: If the drives do not show up in your network tab in file explorer insert the path into the top bar, for example: **\\Server11**

**CONTACT ME AT THIS POINT AND DEMO YOUR SETUP. YOU WILL NEED A SIGN OFF ON THIS.**

- Map the X, Y and Z drives to your local computer. To do this, in File Explorer right-click **This PC** → **Map Network Drive**. You should see a window like the one below:



- Select a corresponding drive letter for each share and use **Browse** to locate it. Make sure **Reconnect at logon** is selected so that you do not have to do this every time you logon to the workstation. Click **Finish**.

**CONTACT ME AT THIS POINT AND DEMO YOUR SETUP. YOU WILL NEED A SIGN OFF ON THIS.**

- Lab Exercise:** Provide a new user, Leslie Knope, with an area on your file server where she can store files and folders for each individual, each client team, and the server team. Those corresponding users should be able to browse and read the corresponding files, while Leslie should have the ability to browse, read, modify, and add files to these directories. Implement this and, in your report, draw the resulting directory structure and the corresponding NTFS and shared permissions.

Hints: Allocate new space on the disk for K:, create the KnopeL user, give her “Full Control” (also called `chmod 777`) rights. Set up the file structure so that KnopeL will be able to edit, create and delete data from corresponding team and individual folders. Individual folders and team folders should have the same permissions as the previous exercise with regards to read options, however with no write access.

**CONTACT ME AT THIS POINT AND DEMO YOUR SETUP. YOU WILL NEED A SIGN OFF ON THIS.**

Before you leave, save your MMC setting, close all applications, and shut down the server and your workstations.

*1 sheet per team – print this sheet and bring it to the lab.*

Your Names: \_\_\_\_\_

Server/Team #: \_\_\_\_\_

Date: \_\_\_\_\_

Sign-offs:

***Disk setup***

***Login from workstation and modify drives, folders, files***

***Map network drives***

***Shared area on file server that users can access***

Your Names: \_\_\_\_\_

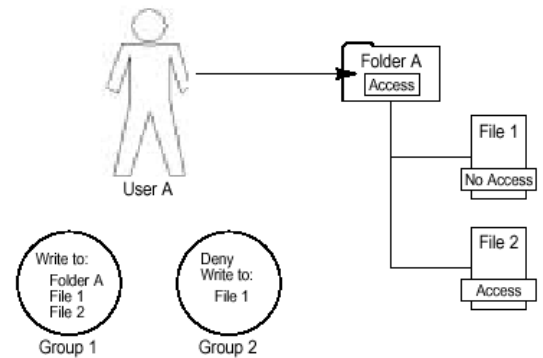
Server/Team #: \_\_\_\_\_

Date: \_\_\_\_\_

### Lab Exercise

Provide a diagram showing the directory structure and the corresponding NTFS and shared permissions you developed.

**Figure 1**  
Deny permission overrides all other permissions



**Figure 2**  
NTFS permission inheritance

