File Sharing Technical Overview

One of the main purposes of networks is to make data available to users in disparate locations. One of the most common types of data to share is a file of some type, and one of the primary mechanisms for making these files available is file sharing.

Note Although the term is *file sharing*, the shared files are actually accessed through shared *folders*.

There are several ways to share a folder, such as the Computer Management console, folder Properties, and even the Configure Your Server wizard that launches the Share a Folder Wizard when configuring a file server. The easiest method especially for the new administrator is to simply run the Share a Folder Wizard. This walks you through creating the share and assigning it permissions.

Note There is also a new Manage Your Server Console that enables you to configure a number of server configuration settings, including shared folders.

File sharing is simply a remote user connecting to a shared folder, either by mapping a drive or directly via universal naming convention (UNC). For example, a share called `software` on the `LONDON` server would be accessed via `\\LONDON\software`. (Servers can also be accessed in the same manner via their DNS style names, such as `\\London.braincore.net\software`.)

Each shared folder is assigned permissions to designate who is allowed to access the share and what level of access they have. The three levels of access for shared folders are Read, Change, and Full Control. In previous versions, all shared folders defaulted to Everyone - Full Control. With Windows Server 2003, however, the default in the wizard is to allow Everyone Read access and to allow Administrators Full Control.

Any folder can be shared regardless of the underlying file system. Folders on FAT and FAT32 partitions can be shared as well as folders on NTFS permissions. It is generally recommended that all partitions (or volumes in the case of dynamic disks) be NT File System (NTFS). The NTFS file system has several advantages over and above FAT and FAT32: better sector allocation, file system security, quota management, compression and encryption capabilities, and so on. Because of the built-in security on NTFS files and folders, you should pay careful attention when sharing these folders.

How Permissions Work

At both the share level and the file level, each resource has certain permissions associated with it. These permissions are maintained in what is called a *Discretionary Access Control List (DACL)*. The DACL is simply a list of *Security Identifiers* (SIDs) and the corresponding permission associated with each SID. Every security principle (user, group, computer, and so on) has its own SID. So, the DACL is simply the list of who has what level of access to the resource. When a user logs on to a domain, he is authenticated via the domain controller. The domain controller determines of which groups the user is a member and builds a token that contains the user's SID and the SIDs of all groups of which the user is a member.

Note This is actually an oversimplification of the process, but it will suffice for our purposes here. This is also why you need to log on and log off to gain the permissions associated with a new group. The SID for the new group isn't in your token until you sign on again.

When the user attempts to access a resource (a file share), this user token is presented to the resource. The operating system then examines the token against the DACL on the resource and compares the SIDs in the user's token against the SIDs in the DACL. For each SID in common, the user has the level of access associated with that SID. For example, let's say the permissions for the `\\LONDON\software` share are Administrators - Full Control, Software Admins - Change, and Domain Users - Read. The actual DACL stored on the shared folder object is the SID for the Administrators group (S-1-5-21-1644491937-1682526488-1202660629-500) - Full Control, the SID for the Software Admins group (S-1-5-21-1644491937-1682526488-1202660629-1008) - Change, and the SID for the Domain Users group (S-1-5-21-1644491937-1682526488-1202660629-513) - Read.

Now let's say the user Mary is a member of the Domain Users group. Mary's SID is S-1-5-21-1644491937-1682526488-1202660629-1005, and the Domain Users group's SID is S-1-5-21-1644491937-1682526488-1202660629-513. When Mary attempts to access the shared folder, her token (which has her SID and the SID of the Domain Users group) is compared to the DACL on the share. Because there is a SID in common (the Domain Users group), she is granted Read access (because that is the level of access granted to the Domain Users group). If a user is a member of multiple groups, the user gets the permissions associated with all those groups.

For example, say Bill is a member of the Software Admins group and the Domain Users group; his effective permissions therefore are Change and Read (Change by virtue of being in the Software Admins group and Read by virtue of being in the Domain Users group).

Windows 2000 provides two check boxes for each permission, labeled Allow and Deny. So far, we have been assuming that we are granting the allow permission. However, you can also explicitly specify a user as having deny permission. A deny permission overrides an allow—this is usually used only to prevent a member of a group from having the permission associated with that group.

For example, let's say you didn't want Bill to have Change permissions to the share anymore, but he still needs to be in the Software Admins group. You could explicitly deny his user account Change permission to the share. Then, he would be back to only Read: Domain Users - Read; Software Admins - Change; and Bill's user account - deny Change, which overrides the allow change granted by the Software Admins group and leaves him with only Read.

I mentioned before that different types of objects have different permissions and that NTFS drives have security. On NTFS drives, you can grant permissions to files and folders just like you can shared folders. The actual permissions you can assign are more numerous, though. For example, on an NTFS folder you can grant (or deny) Read, Write, List Folder Contents, Read and Execute, Modify, or Full Control.

Note Actually, each of these permission is composed of several more granular permissions—for example, Read is actually List Folder/Read Data, Read Attributes, Read Extended Attributes, and Read Permission.

NTFS and Shared Folder Cumulative Permissions

If you specify permissions to an NTFS file or folder, you are again putting entries in a DACL. As before, when a user attempts to access that file or folder, the user's token is compared to the DACL. The user is then allowed access based on the matches between the SIDs on the DACL and the SIDs in the user's token.

When an NTFS folder is shared and accessed across the network, two objects are involved: the file share and the NTFS file or folder. The common description of the effective permissions is that the most restrictive of the two is applied. This can be explained by thinking about what is actually happening. Accessing an NTFS file or folder through a network share is basically a two-step process. First, the user accesses the file share (`\\LONDON\software`); then she accesses the file or folder (`software` or `word.exe`). When accessing the file share, the user's token is compared to the DACL on the file share and she gets whatever level of access is granted by the share (Full Control, Change, or Read). Regardless of which permissions are on the NTFS file or folder, the user can't do anything through the share that her share permissions won't allow. In the example of Mary, she has Read permissions to the share, so even if she has Full Control to the file, she can still only read through the share. The second step is the accessing of the file or folder. Once again, the user's token is compared to the DACL and the user gets that level of access, but only as much access as the DACL specifies. Let's say Bill has Change and Read permission to the share but is allowed only Read and Execute permission to the file. Even though he has Change permission at the share, he would not be able to modify the file because he doesn't have write permission to the file.

Caution To prevent confusion between accessing a file or folder locally (where only the NTFS permissions apply) and accessing it across the network (where both share and NTFS permissions apply), previous versions of Windows default to allowing Everyone Full Control permissions to the share. Windows Server 2003 is much more security conscious, so the default setting for file shares is to allow Administrators Full Control and Everyone Read (although you can still choose Everyone Full Control in the wizard). If you do allow Everyone Full Control to the share, be very sure that more restrictive permissions exist at the file and folder levels.