

Automatically Granting NTFS Permissions Based on Folder Names in Active Directory

Managing folder permissions in a Windows environment can be a tedious task, especially when you have a large number of users and corresponding directories. Imagine a scenario where every user has their own folder, and you want to automatically give them the right NTFS permissions without manually clicking through every folder's properties. That's where PowerShell shines.

With a simple script, you can loop through a set of folders, match each folder name to an Active Directory (AD) username, and automatically grant the appropriate permissions. The beauty of this approach is that it scales effortlessly, reduces human error, and keeps your file system secure and organized.

Here's an example script that does exactly that:

```
# Define the path where the folders are
$FolderPath = "C:\SharedFolders"

# Define the permission to grant
$Permission = "Modify"

# Loop through each folder
Get-ChildItem -Path $FolderPath -Directory | ForEach-Object {
    $FolderName = $_.Name
    $FolderFullPath = $_.FullName

    # Check if a user with that username exists in Active Directory
    $ADUser = Get-ADUser -Filter { SamAccountName -eq $FolderName }

    if ($ADUser) {
        Write-Host "Granting $Permission permission to $FolderName on $FolderFullPath"

        # Grant NTFS permissions
        $acl = Get-Acl $FolderFullPath
        $accessRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
            $ADUser.SamAccountName,
```

```
        $Permission,  
        "ContainerInherit,ObjectInherit",  
        "None",  
        "Allow"  
    )  
    $acl.SetAccessRule($accessRule)  
    Set-Acl -Path $FolderFullPath -AclObject $acl  
} else {  
    Write-Warning "No AD user found matching folder name: $FolderName"  
}  
}
```

Revision #1

Created 17 September 2025 07:38:08 by Luke Carter

Updated 17 September 2025 07:38:51 by Luke Carter