

Active Directory

Powershell Scripts for Active Directory

- [Import & Export AD Passwords](#)
- [Updating AD User extensionAttribute10 for 365 Address Lists](#)

Import & Export AD Passwords

Export AD Usernames with Password Hash

This Export script pulls a list of Active Directory usernames along with their NT password hashes from your domain. It saves them into a CSV file, which you can use later for things like backups, migrations, or syncing accounts across environments. Just pop in your domain name and you're good to go!

On line 1, replace <DOMAIN NAME> with your Active Directory domain name. Example: school.internal

```
Get-ADReplAccount -All -Server "<DOMAIN NAME>" |  
Select-Object SamAccountName, NTHash |  
Export-Csv -Path "C:\Temp\ADExport.csv" -NoTypeInfo
```

Import AD Password Hashes

This Import script takes that exported CSV and restores the password hashes back into Active Directory. It checks each user exists first, then applies the original password hash. Just make sure to update the domain name, and the script handles the rest. Handy if you have migrated users between different Active Directory domains and you want users to keep their same passwords.

On line 26, replace <DOMAIN NETBIOS NAME> to your Active Directory NetBIOS name. Example: school (without .internal)

```
# Path to your CSV file  
$csvPath = "C:\Temp\ADExport.csv"  
  
# Import the CSV  
$users = Import-Csv -Path $csvPath  
  
foreach ($user in $users) {  
    $sam = $user.SAMAccountName  
    $hash = $user.NTHash  
  
    # Skip if hash is empty  
    if ([string]::IsNullOrEmpty($hash)) {  
        Write-Verbose ("Skipping $sam (no NTHash)") -Verbose  
    }  
}
```

```
        continue
    }

    # Check if account exists
    $exists = Get-ADUser -Filter { SamAccountName -eq $sam } -ErrorAction SilentlyContinue
    if ($null -eq $exists) {
        Write-Verbose ("Skipping $sam (account not found)") -Verbose
        continue
    }

    # Set the password hash
    try {
        Set-SamAccountPasswordHash -SamAccountName $sam -domain <DOMAIN NETBIOS NAME> -NTHash
$hash -Verbose
    } catch {
        $errMsg = $_.Exception.Message
        Write-Verbose -Message ("Failed to set hash for user '$sam'. Error: $errMsg") -Verbose
    }
}
```

Updating AD User extensionAttribute10 for 365 Address Lists

This script grabs all users in a specific OU, then updates their extensionAttribute10 to whatever value you want—like the name of the address list they should belong to. Just point it at your OU, set the value, and run. Each user gets updated, and you get a confirmation message for every change.

It's a simple way to bulk-assign users to the correct address lists without clicking through each AD user one by one. Saves time and keeps your GAL organized!

```
# Import Active Directory module
Import-Module ActiveDirectory

# Define the value to be added and the attribute name
$attributeValue = "Address List Name"
$attributeName = "extensionAttribute10"

# Specify the OU (Organizational Unit) - replace with your OU's distinguished name
$ouDistinguishedName = "OU=*****,OU=*****Users,OU=*****,OU=*****,
DC=*****,DC=*****,DC=**,DC=**"

# Get all users in the specified OU
$usersInOU = Get-ADUser -Filter * -SearchBase $ouDistinguishedName
# Loop through each user and update the attribute
foreach ($user in $usersInOU) {
    Set-ADObject -Identity $user -Add @{ $attributeName = $attributeValue }
    Write-Host "Attribute $attributeName updated for $($user.SamAccountName)"
}

# Output a final confirmation message
Write-Host "All users in $ouDistinguishedName have been updated."
```