Active Directory

- Export users with home directory set to share
- Set user home to path or local home
- Report, export & manage user logon scriptpath

Export users with home directory set to share

This PowerShell script simplifies Active Directory user management. It quickly identifies users with network share home directory paths within a specified OU. You can easily view their details, such as names, account names, and paths, and export this information to timestamped CSV files. It's a practical tool for maintaining an organized AD environment.

```
# Get all AD users within the specified OU
$users = Get-ADUser -Filter * -Properties HomeDirectory

# Define an empty array to store users without a local path for the Home folder
$ADUsersWithPath = @()

# Loop through each user and check if the HomeDirectory property is not a local path
foreach ($user in $users) {
    if ($user.HomeDirectory -like "\\*") {
        $ADUsersWithPath += $user
    }
}

# Display the list of users without a local path for the Home folder
$ADUsersWithPath | Select-Object Name, SamAccountName, HomeDirectory
$ADUsersWithPath | Export-CSV ".\$(Get-Date -Format yyMMdd) AdUsersWithHomeDirectoryPath.csv"
```

Set user home to path or local home

This PowerShell script streamlines the process of setting or updating home directory paths for multiple Active Directory (AD) users. It reads a list of usernames or User Principal Names (UPNs) from a file and uses the Set-ADUser cmdlet for the task. You can customize the home directory path or leave it empty to retain existing paths. Ideal for administrators managing users in a specific OU, the script enhances efficiency and maintains consistency within an AD environment.

```
# Get all AD users within the specified OU
$users = Get-Content -Path "<yourfilepathwithusernamesorupns>"

# Set the pathVariable or leave empty for local home path
$homePath = ""

# Loop through each user and check if the HomeDirectory property is not a local path
foreach ($user in $users) {
    Set-ADUser -Identity $user -HomeDirectory $homePath
}
```

Report, export & manage user logon scriptpath

> These script blocks must be run on a domain controller/domain computer and with permissions to read or modify domain users.

In the Active Directory, scripts can be set on user objects that are executed on logon. These scripts must be stored in "NETLOGON" so that they can be added with their file name.
Since these scripts somewhat restrict the visibility of functions and dynamics, it is better to use group policy objects (GPOs).

This guide is about how to evaluate and remove these scripts for all user objects.

Get overview

The first query returns all the scripts used, so you can get an overview of which scripts are actually being used.

```
(Get-ADUser -Filter * -Property * | select ScriptPath).ScriptPath | Sort-Object | Get-Unique
```

Report user by script path

All users who use a specific script can be displayed using this query. This allows the individual objects to be adjusted after evaluation during a migration.

```
$ScriptPath = "<yourscriptpath>"
((Get-ADUser -Filter * -Property *) | where { $_.scriptpath -contains $ScriptPath}) | % {
    Write-Output "Path set for: $($_.UserPrincipalName)"
}
```

Remove specific entries

This PowerShell snippet can be used to remove the logon script on the user objects that have a special path set. In addition, the manipulated objects are output to the console.

```
$ScriptPaths = @(
  "<yourscriptpath1>",
  "<yourscriptpath2>"
)

foreach($ScriptPath in $ScriptPaths){
    # ((Get-ADUser -Filter * -Property *) | where { $_.scriptpath -contains $ScriptPath}).count
    ((Get-ADUser -Filter * -Property *) | where { $_.scriptpath -contains $ScriptPath}) | % {
        Set-ADUser $_.samaccountname  -Clear ScriptPath
        Write-Output "Cleared ScriptPath $ScriptPath for $($_.UserPrincipalName)"
    }
}
```