



A3 Växel

AD integration



För en automatisering behövs en mapp som A3-växeln kan nå och har rättighet att läsa. Sedan behövs ett automatiserat script som läser ut användarinformation från AD'et tillsammans med växel användaren.

Vi har två PowerShell script som går att använda, det ena som läser in växel användaren. De andra som läser ut användarinformationen.

Input

Inläsningen av växel användarna använder sig av en komma (,) separerad csv-fil som ska innehålla AD'ts **samAccountName** och **växel användaren**. Csv-filen ska på första raden innehålla rubrikerna *samAccountName* och *Phone*. Rubriken och varje AD-användare ska vara på en egen rad enligt exemplet nedan.

```
samAccountName,Phone  
adUser1,vxlUsr123456  
adUser2,vxlUsr345678  
OSV.
```

Du bör ha PowerShell version 5 eller högre installerat för att använda dessa script. Scripten innehåller lite olika alternativ beroende på om du kör scripten från AD-servern eller från en annan dator i nätverket.

```
$CSV Filepath = 'my importfile and path'  
$usrList = Import-Csv $CSV_Filepath  
  
foreach($usr in $usrList){  
    #To write a output of the imported csv-file (remove # to activate the code)  
    #Write-Host $usr.SamAccountName $usr.Phone  
  
    #Code to write phone nr to all users in the csv-file (remove # to activate the code)  
    #Set-ADUser -Identity $usr.SamAccountName -replace @{homephone= $($usr.Phone)}  
}
```

För att inte riskera att köra scriptet oavsiktligt har kod-raden som skriver in informationen från csv-filen till AD't kommenterats bort med hjälp av #.

Du kan testköra scriptet och se att du får rätt information från importfilen genom att aktivera kod-raden *Write-Host*, ta bort # framför raden och kör scriptet. Informationen kommer att skrivas ut på skärmen.

I detta exempel av script har fältet **HomePhone** används, men de finns fyra andra fält att använda på AD kortets telefon tab. Dessa finns i scriptfilen tillsammans med exempel på kod för att köra skriptet från en annan dator än AD-servern.

Output

Detta script används för att läsa ut användarinformation ur AD't tillsammans med den växel användaren som lagts till varje AD-användare av input-scriptet. De kommer att hålla användarinformation och växel uppdaterad. PowerShell-scriptet skapar en kommaseparerad textfil som placeras där A3-växeln kommer åt den. Scriptet innehåller några parametrar som krävs för att allt ska fungera.





Första parametern `$outputfilepath` ska innehålla sökväg och filnamn till den fil som växeln ska läsa. Den andra `$title` är en titel för att växeln ska kunna sortera informationen. Tredje är `$phonefield` som måste vara densamma som användes vid input-scriptet, de finns fem olika värden som går att använda. Namn och information om hur man ändrar finns beskrivet i script-filen. Fyll i dina värden enligt exemplet nedan.

```
$outputfilepath = '\\Your\shared\filelocation\a3v.txt'  
$title = 'Youre title'  
$phonefield = 'mobilephone'
```

Det finns olika metoder att få scriptet att köras automatiskt i ett önskat intervall, vi använder oss av Windows egna schemaläggaren.

1. Spara scriptet i en egen mapp ex. `c:\script`
2. Skapa en ny aktivitet, fyll i ett namn.
Klicka på **Byt användare eller grupp**
den användaren du valt måste ha explicita rättigheter i mappen där scriptet är sparat och där filen från scriptet ska sparas. Den måste även ha rättighet att köra scriptet på AD servern.
3. Klicka i **Kör med högsta privilegier**
4. Under fliken *Utlösare*, fyll i uppgifter om när och hur ofta du vill att scriptet ska köras.
5. Under fliken *Åtgärder*, välj **Starta ett program**
I *Program eller script*, fyll i **powershell**
I *Lägg till argument*, fyll i **-file c:\script\A3PS_Get_ADUserInfo.ps1**
I *Starta i*, fyll i **c:\script**

Spara det nya schemat och prova att köra för att se dina inställningar tillåter scriptet att köra och att rättigheterna är tillräcklig för att scriptet får spara filen där du valt.

OBS!

Om du vill att scriptet är automatiserat och att de körs från en annan maskin än AD't. Måste en fil med användaruppgifter och lösenord skapas och skickas med när scriptet körs. Ett exempel på detta ser du nedan.

```
$passwordPath = './password.txt'  
if (-not (test-path $passwordPath)) {  
    $cred = Get-Credential -Username domain\username -message 'Please login.'  
    Export-Clixml -InputObject $cred -Path $passwordPath  
}  
$cred = Import-CliXML -path $passwordPath  
  
$session = New-PSSession -ComputerName ad-server.domän.local -Credential $cred
```

