

DHCP Sunucusundan IP Silme

IP bloklarının verebildiđi IP adres sayılarının yetersiz olduđu durumda bazı IP adreslerinin DHCP'den silinmesi gerekmektedir. Adresleri tek tek silinebileceđi gibi bir powershell scripti ile toplu da silinebilir. Ařađıda yer alan [powershell](#) scripti kullanılarak belli řartları řađlayan IP adreslerini kolaylıkla sileriz. Script [active directory](#) kullanıcısının yetkisine sahip olması durumunda kendi bilgisayarında da alıřtırabilir.

Ařađıdaki örnek scripte isminde android, galaxy veya phone geen ve rezerve olmayan IP adresleri siliniyor.

PowerShell İle İşletim Sistemine Göre AD Bilgisayar Sayılarını Bulma

Count	Name
69	Windows 10 Pro
5	Windows 10 Pro for Workstations
158	Windows 7 Professional
5	Windows 8 Pro
42	Windows 8.1 Pro
1	Windows Server 2003
7	Windows Server 2008 R2 Enterprise
1	Windows Server 2008 R2 Standard
3	Windows Server 2012 Datacenter
27	Windows Server 2012 R2 Standard
6	Windows Server 2012 Standard
9	Windows XP Professional

Merhaba arkadaşlar,

[Sistem yöneticilerinin](#) özellikle lisans denetimleri öncesinde ve anti virüs yazılımları gibi bütün makinelere kurulacak sistemler öncesi saha analizinde işletim sistemlerine göre bilgisayar sayıları ihtiyaç duydukları hayati bilgilerden biri olabiliyor. AD'ye ilişkin hemen hemen her soruya cevap veren [PowerShell](#) ile bunun cevabını vermek de son derece basittir. Aşağıdaki powershell scripti ile yukarıdaki ekran görüntüsünde görüldüğü gibi bilgisayar sayıları kolaylıkla elde edilebilir.

```
# Import AD module
Import-Module ActiveDirectory
```

```
# Domain adını bulma
$DomainName = (Get-ADDomain).NetBIOSName
```

```
# Kaç gün öncesine kadar oturum açmış makineleri sorgula
$days = 30
$lastLogonDate = (Get-Date).AddDays(-$days).ToFileTime()
```

```
# AD sorgulama
$Computers = @(Get-ADComputer -Properties
Name,operatingSystem,lastLogonTimeStamp -Filter
{(OperatingSystem -like "*Windows*") -AND (lastLogonTimeStamp
-ge $lastLogonDate)})
foreach($Computer in $Computers)
{
    $Computer.OperatingSystem = $Computer.OperatingSystem -
replace '®' -replace '™' -replace '□□□','Professional (Ch)' -
replace 'Professionnel','Professional (Fr)'
}

$Computers | Group-Object operatingSystem | Select Count,Name
| Sort Name | Out-GridView
```

Uzaktan DNS Adresi Değiştirme



Sistem yöneticileri için network yapısı ve içerisinde yer alan cihazların sürekliliği ön plandadır. Bazı durumlarda uzakta yer alan cihazların (Windows için) DNS ayarlarının değiştirilmesi gerekmektedir. Burada devreye Microsoft'un mükemmel denebilecek ürünlerinden biri olan Powershell devreye girmektedir.

Aşağıda yer alan powershell scriptini kullanarak uzak bilgisayarın dns ayarlarını değiştirebilirsiniz.

- İlk satırda yer alan “C:\list\servers.txt” dosyası dns ayarları değiştirilecek makinelerin IP adresleri veya isimleri yer alıyor. Dosya yapısı her satıra bir makine bilgisi gelecek şekilde düzenlenmelidir.
- \$DNSServers = “10.1.0.254”, “10.1.0.253” satırında da verilecek yeni DNS IP adresleri veriliyor. Bu örneğimizde 254 IP adresi birinci DNS, 253 IP adresi de ikinci DNS olarak ayarlanacak.

```
$computer = get-content C:\list\servers.txt
```

```
$NICs = Get-WMIObject Win32_NetworkAdapterConfiguration -  
computername $computer |where{$_.IPEnabled -eq “TRUE”}
```

```
    Foreach($NIC in $NICs) {
```

```
$DNSServers = “10.1.0.254”, “10.1.0.253”
```

```
    $NIC.SetDNSServerSearchOrder($DNSServers)
```

```
    $NIC.SetDynamicDNSRegistration(“TRUE”)
```

```
}
```

Sistemde Oturum Açan Hesabın Sorgulanması

Sistem yöneticilerinin, sistemi yönetirken bilmeleri gereken bilgilerden biri de hangi bilgisayarda hangi hesabın oturum açtığı bilgisidir. Bu bilgiye basit bir [powershell](#) komutu ile ulaşabiliriz.



Sorguyu yapmak için kullanabileceğimiz [powershell](#) sorgusu:

Organization Unit Yapısını Başka Bir Domain'e Kopyalama

[Active Directory](#) ile uğraşan arkadaşlar bazı sebeplerden dolayı OU yapısını başka bir sunucuya kopyalamak isteyebilirler. Bu işlem için aşağıdaki [powershell](#) kodları kullanılabilir. Kodların çalışması için [powershelle](#) activedirectory modülünün tanıtılmış olması gerekmektedir.

Öncelikle aşağıdaki kod ile mevcut yapı bir txt dosyasına alınır.

Burada "OU=LAB" ifadesinde LAB yerine dışarı aktarılacak OU adı, DC ifadelerinde de LAB yerine domain adı, SE yerine de domain uzantınızı yazınız.

Daha sonra da hedef domainde aşağıdaki kod çalıştırılarak OU yapısı taşınmış olur.

Windows 10 Kaldırılmayan Uygulamaları Kaldırma

Windows 10 ile gelen varsayılan programlar faydalı olsalar da bazıları da varlıkları ile bile sinir bozucu. Windows 10 ile gelen ve varsayılan olarak kaldırılmayan uygulamaları kaldırmak için uygulanacak adımlar aşağıdaki gibidir:

1. **Başlat** menüsünde **Tüm Programlar > Windows Powershell** komut sistemini sağ tıklayarak yönetici olarak çalıştır tıklayınız. Açılan onay penceresinde **Evet** tıklayarak onaylayınız.
2. Açılan PowerShell ekranına aşağıdaki kodu yazın
3. **Enter** tuşuna basarak kodu çalıştırın. İşlem sonunda Windows 10 ile gelen hesap makinesi kaldırılacaktır.

Başka programları kaldırmak için **windowscalculator** yerine aşağıda program karşılıklarında yazan isimlerini yazıp aynı adımları tekrarlayın.

3D Builder: **3dbuilder**

Alarms and Clock (Alarm ve Saat): **windowsalarms**

Calendar and Mail (Takvim ve Mail): **windowscommunicationsapps**

Camera (Kamera): **windowscamera**

Groove Music: **zunemusic**

Maps (Haritalar): **windowsmaps**

Movies & TV (Filmler ve TV): **zunevideo**

OneNote: **onenote**

People: **people**

Photos (Kişiler): **photos**

Store (Mağazalar): **windowsstore**

Voice Recorder (Ses Kaydı): **soundrecorder**

Xbox: **xboxapp**

Uzak Bilgisayarda Aktif Olan Hesabı Bulma

❌ Domain yapısında bulunan bilgisayarlarda bazı durumlarda uzakta bulunan bir bilgisayarda anlık hangi kullanıcının oturum açtığı sorgulanmak istenebilir. Bu durumda powershell komut sisteminde aşağıdaki script çalıştırılır:

Script çalıştırıldıktan sonra öncelikle sorgulanacak bilgisayar adı istenecek, bilgisayar adı girilip Enter basıldıktan sonra bilgisayarda açık olan hesap görüntülenecektir.

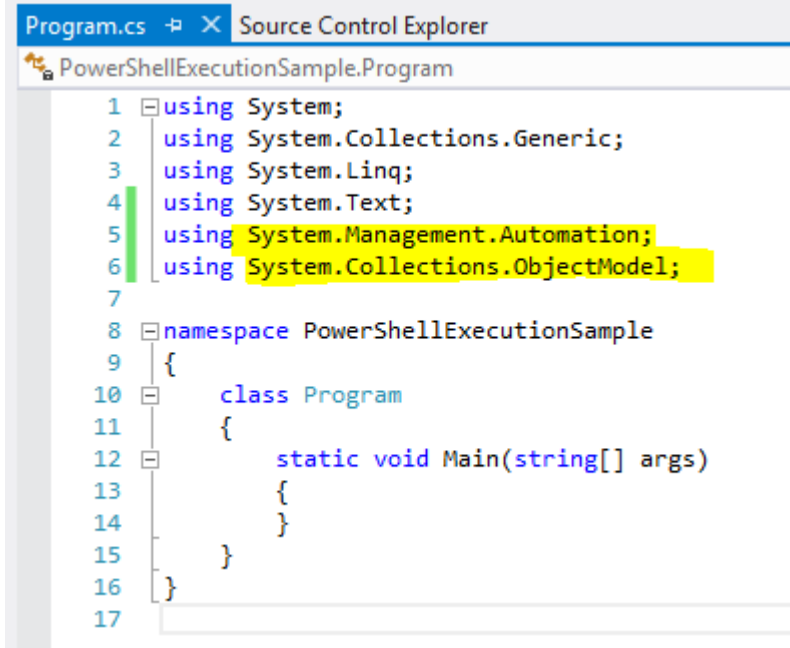
Powershell Komutlarının C# ile Çalıştırılması

Windows ortamının en kullanışlı araçlarının başında Powershell komutları gelmektedir. Bu araç sayesinde bir çok işlem kolayca yapılabilmektedir. Powershell komutlarının C# içerisinde kullanılması da aynı şekilde zorlu birçok işlemin kolayca

yapılmasını sağlar.

Şimdi adımları yazalım:

- ***System.Management.Automation*** referansını projemize ekliyoruz.
- “***System.Collections.ObjectModel***” ve “***System.Management.Automation***” referanslarını using ifadesi ile sayfaya ekliyoruz.



```
Program.cs Source Control Explorer
PowerShellExecutionSample.Program
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Management.Automation;
6 using System.Collections.ObjectModel;
7
8 namespace PowerShellExecutionSample
9 {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14         }
15     }
16 }
17
```

- Powershell işlemini yapacak nesne oluşturulur.
- Oluşturulan powershell nesnesine komut ve parametreler eklenir.
- Powershell komutunun çalıştırılması

Örnek çalışma:

Uzak Bilgisayar Paylaşılan Klasörleri Bulma

Powershell komutu ile uzaktaki herhangi bir Windows yüklü bilgisayarda paylaşıma açılmış klasörleri görüntüleyebiliriz. Bunun için gerekli olan kod parçası.

Aynı şekilde bu powershell komutunu C# ile çalıştırarak da bir arayüz üzerinden sonuçlar görüntülenebilir. Bunun için gerekli olan C# kodu aşağıdadır. Kod içerisinde kullanılan Shared isimli sınıfımızın kodu:

Powershell ile Yazıcı Listesini Dosyaya Alma

Bilgisayarımıza tanımlı yazıcıların listesini .csv dosyasına almak için powershell komut yorumlayıcısını açarak aşağıdaki kodu çalıştıralım:

