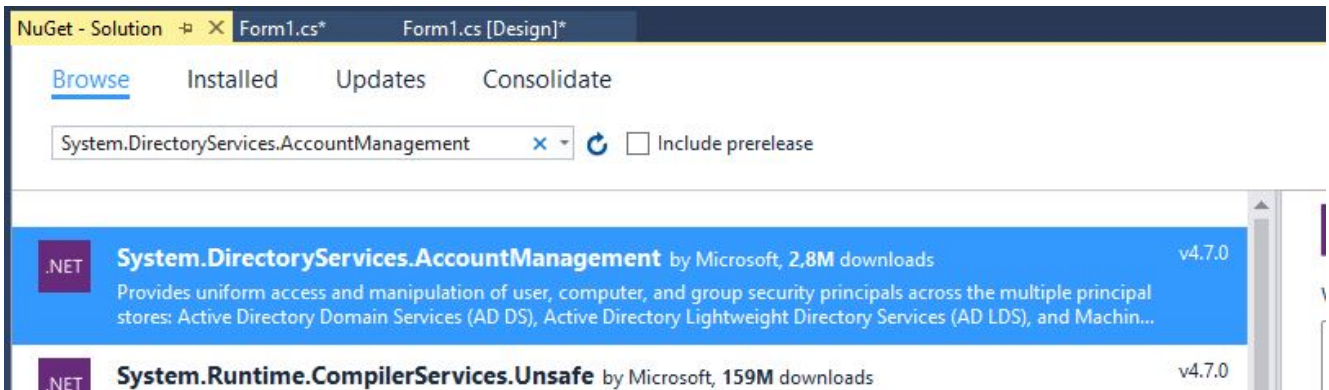


# Sistem Yöneticileri İçin C# Kodları

Bilgisayar biliminin yaygın alanlarından ikisi sistem yönetimi ve yazılım alanlarıdır. Her ne kadar bu iki alanda çalışan arkadaşlar bir birine karışmamaya çalışsa da tecrübelerimiz bize farklı şeyler söylüyor. Tecrübelerimiz, her sistem yöneticisi arkadaşın az da olsa yazılım, her yazılımcı arkadaşın da az da olsa sistem yönetimi konusunda tecrübe edinmesi gerektiğini söylüyor. Bu yazımızda sistem yönetimi için kullanılan bazı kodları paylaşıyor olacağız.

## Active Directory Kullanıcı ve Grup İşlemleri

AD işlemleri için .Net Framework 4.5 ile gelen System.DirectoryServices.AccountManagement isim uzayında (Namespace) yer alan sınıfları kullanacağız. İlgili paketi Nuget'ten projemize ekledikten sonra işlemlere başlayabiliriz. Paket [linki:](https://www.nuget.org/packages/System.DirectoryServices.AccountManagement/)  
<https://www.nuget.org/packages/System.DirectoryServices.AccountManagement/>



Active Directory işlemlerinin tamamında sunucu ile bağlantıyı sağlayan `PrincipalContext` sınıfından nesne türetilmelidir. `PrincipalContext` sınıfının yedi farklı oluşturucu metodu olsa da en sık kullanılan üç metodu şunlardır;

- `PrincipalContext(ContextType)`: Sadece bağlanacak sunucun türünün alındığı oluşturucu metottur. `ContextType` enum değerinin alabileceği değerler:
  - \* `ApplicationDirectory`: Uygulama dizin sunucusu
  - \* `Domain`: Domain controllerdan verileri okumak için okunur.
  - \* `Machine`: Local makine SAM veri tabanına bağlantıda kullanılır.
- `PrincipalContext(ContextType, String)`: İlkinin yanında ikinci parametre olarak bağlanacak domain controller adını alır. `cozumpark.local` gibi...
- `PrincipalContext(ContextType, String, String, String)`: İlk iki metodun parametrelerinin yanında üçüncü parametre olarak bağlantıda kullanılacak kullanıcı adı ve dördüncü parametre olarak da girilen kullanıcı adının şifresini alır.

## **AD Kullanıcılarının Sistemden Alınması**

AD kullanıcılarının sistemden alınabilmesi için oluşturulan `PrincipalContext` nesnesini parametre alan bir `UserPrincipal` nesnesi oluşturulur. Daha sonra bu nesne üzerinden istenen filtre değerleri girilir. Örneğin aşağıdaki kodda sadece aktif kullanıcıları istediğimizden `"userPrincipal.Enabled = true;"` ataması ile bunun filtreye ekliyoruz. Daha sonra oluşturulan bu `UserPrincipal` nesnesini parametre alan bir `PrincipalSearcher` nesnesi oluşturulur. Son adım olarak `PrincipalSearcher` nesnesinin `FindAll` metodu ile `UserPrincipal` üzerinden girilen değerlere uygun kullanıcılar sistemden

alınır. Metotdan dönen değerler PrincipalSearchResult türünde olduğundan "Cast<UserPrincipal>()" metodu ile dönen sonuç UserPrincipal türüne çevrilir.

```
PrincipalContext principalContext = new
PrincipalContext(ContextType.Domain, "saitorhan.local",
"Administrator", "123456qaZ.");
UserPrincipal userPrincipal = new
UserPrincipal(principalContext);
userPrincipal.Enabled = true;
PrincipalSearcher principalSearcher = new
PrincipalSearcher(userPrincipal);
List<UserPrincipal> principalSearchResult =
principalSearcher.FindAll().Cast<UserPrincipal>().OrderBy(u =>
u.SamAccountName).ToList();
```

[1]		
	Name ("Sait ORHAN")	System.DirectoryServices.AccountManagemen
AccountExpirationDate	null	System.DateTime?
AccountLockoutTime	null	System.DateTime?
AdvancedSearchFilter	{System.DirectoryServices.AccountManagement.AdvancedFilters}	System.DirectoryServices.AccountManagemen
AllowReversiblePasswordEncryption	false	bool
BadLogonCount	0	int
Certificates	{System.Security.Cryptography.X509Certificates.X509Certificate2Collectio	System.Security.Cryptography.X509Certificat
Context	{System.DirectoryServices.AccountManagement.PrincipalContext}	System.DirectoryServices.AccountManagemen
ContextRaw	{System.DirectoryServices.AccountManagement.PrincipalContext}	System.DirectoryServices.AccountManagemen
ContextType	Domain	System.DirectoryServices.AccountManagemen
DelegationPermitted	true	bool
Description	null	string
DisplayName	"Sait ORHAN"	string
DistinguishedName	"CN=Sait ORHAN,OU=BilgiTeknolojileri,OU=Users,OU=..."	string
EmailAddress	"saitorhan@..."	string
EmployeeId	null	string
Enabled	true	bool?
GivenName	"Sait"	string
Guid	{ac788b01-d395-4208-8e00-1bedcfa0db2d}	System.Guid?
HomeDirectory	null	string
HomeDrive	null	string
LastBadPasswordAttempt	{5.02.2020 09:13:02}	System.DateTime?
LastLogon	{5.02.2020 00:17:12}	System.DateTime?
LastPasswordSet	{20.01.2020 06:38:33}	System.DateTime?
MiddleName	null	string
Name	"Sait ORHAN"	string
PasswordNeverExpires	true	bool
PasswordNotRequired	false	bool
PermittedLogonTimes	null	byte[]
PermittedWorkstations	{System.DirectoryServices.AccountManagement.PrincipalValueCollection	System.DirectoryServices.AccountManagemen
SamAccountName	...	string
ScriptPath	null	string
Sid	{S-1-5-21-2023297446-2633530542-46019157-4110}	System.Security.Principal.SecurityIdentifier
SmartcardLogonRequired	false	bool
StructuralObjectClass	"user"	string
Surname	"ORHAN"	string
UserCannotChangePassword	false	bool
UserPrincipalName	...	string
VoiceTelephoneNumber	null	string
Static members		
Non-Public members		

Ekran görüntüsünde de görüldüğü üzere bu şekilde kullanıcı bilgileri AD Sunucusu üzerinden alındığında hesap ile ilgili her türlü bilgi elde edilebiliyor.

## Kullanıcının Üyesi Olduğu Grupları Bulma

Kullanıcının gruplarını bulmak için öncelikle "UserPrincipal.FindByIdentity" metodu ile üyesi olduğu grupların sorgulanacağı kullanıcıyı sistemden alıyoruz. UserPrincipal.FindByIdentity metodu üç parametre alır. Bu parametreler sırasıyla;

1. Yazının başında tanımını yaptığımız PrincipalContext nesnesi
2. Kullanıcı hesabı sorgulanırken kullanılacak özellik. Biz örneğimizde SamAccountName özelliği ile sorgulama yapacağımızdan IdentityType.SamAccountName değerini kullanıyoruz.
3. Sorgulama yapılacak özelliğe ait değer

```
PrincipalContext principalContext = new
PrincipalContext(ContextType.Domain, "saitorhan.local",
"Administrator", "123456qaZ.");
UserPrincipal userPrincipal =
UserPrincipal.FindByIdentity(principalContext,
IdentityType.SamAccountName, "saitorhan");
List<GroupPrincipal> principalSearchResult =
userPrincipal.GetGroups().Cast<GroupPrincipal>().ToList();
```

Dönen UserPrincipal nesnesinin GetGroups metodu ile kullanıcının üyesi olduğu gruplar alınır ve gene Cast<GroupPrincipal>() metodu ile dönen sonucu grupların temsil edildiği GroupPrincipal türüne çeviriyoruz.

principalSearchResult	Count = 19	System.Collections.Generic.List<S
▶ [0]	Name ( "Domain Users" )	System.DirectoryServices.Accountf
▶ [1]	Name ( "Organization Management" )	System.DirectoryServices.Accountf
▶ Context	{System.DirectoryServices.AccountManagement.PrincipalCo	System.DirectoryServices.Accountf
▶ ContextRaw	{System.DirectoryServices.AccountManagement.PrincipalCo	System.DirectoryServices.Accountf
▶ ContextType	Domain	System.DirectoryServices.Accountf
▶ Description	"Members of this management role group have permis Q	string
▶ DisplayName	null	string
▶ DistinguishedName	"CN=Organization Management,OU=Microsoft Exchan Q	string
▶ GroupScope	Universal	System.DirectoryServices.Accountf
▶ Guid	{7471e884-1e6e-46cc-b836-e6c0d7906ea7}	System.Guid?
▶ IsSecurityGroup	true	bool?
▶ Members	{System.DirectoryServices.AccountManagement.PrincipalCo	System.DirectoryServices.Accountf
▶ Count	7	int
▶ IsReadOnly	false	bool
▶ IsSynchronized	false	bool
▶ SyncRoot	{System.DirectoryServices.AccountManagement.PrincipalCo	object {System.DirectoryServices.A
▶ Non-Public members		
▶ Results View	Expanding the Results View will enumerate the IEnumerable	
▶ [0]	Name ( "██████████" )	System.DirectoryServices.Accountf
▶ [1]	Name ( "██████████" )	System.DirectoryServices.Accountf
▶ [2]	Name ( "██████████" )	System.DirectoryServices.Accountf
▶ [3]	Name ( "██████████" )	System.DirectoryServices.Accountf
▶ [4]	Name ( "Sait ORHAN" )	System.DirectoryServices.Accountf
▶ [5]	Name ( "Administrator" )	System.DirectoryServices.Accountf
▶ [6]	Name ( "Exchange Trusted Subsystem" )	System.DirectoryServices.Accountf
▶ Name	"Organization Management"	string
▶ SamAccountName	"Organization Management"	string
▶ Sid	{S-1-5-21-2023297446-2633530542-46019157-1112}	System.Security.Principal.SecurityI
▶ StructuralObjectClass	"group"	string
▶ UserPrincipalName	null	string
▶ Non-Public members		
▶ [2]	Name ( "Help Desk" )	System.DirectoryServices.Accountf
▶ [3]	Name ( "genel_██████████@██████████.com.tr" )	System.DirectoryServices.Accountf
▶ [4]	Name ( "genel_██████████" )	System.DirectoryServices.Accountf
▶ [5]	Name ( "Bilgi Teknolojileri" )	System.DirectoryServices.Accountf
▶ [6]	Name ( "genel_██████████" )	System.DirectoryServices.Accountf
▶ [7]	Name ( "internet_sinirsiz" )	System.DirectoryServices.Accountf
▶ [8]	Name ( "spamtitan" )	System.DirectoryServices.Accountf
▶ [9]	Name ( "Bilgi Teknolojileri" )	System.DirectoryServices.Accountf

Ekran görüntüsünde de görünen enteresan bir duruma dikkatinizi çekmek istiyorum. Sonuç dönen grupların "Members" özelliğinin içinde o gruba üye olan kullanıcıları görebiliyoruz. Tabii bu kullanıcıların içerisinde de üye oldukları gruplar görünüyor. Bu iç içe sonuç kümeleri sayesinde aslında kullanıcının birinin hesabından girip aslında bütün kullanıcı ve grup bilgilerine ulaşılabilir. □

## AD Gruplarının Sistemden Alınması

AD sunucusundan grupların alınması işlemi de yazının ilk





GroupPrincipal.FindByIdentity metodu ile kullanıcı eklenecek grup seçilir. Daha sonra dönen GroupPrincipal türündeki grup nesnesinin Members özelliğine Add metodu (group.Members.Add) ile kullanıcı eklenir. Add metodu üç parametre alır. Bu parametreler;

1. Sunucu ile bağlantıyı sağlayan PrincipalContext
2. Eklenecek kullanıcının hangi özelliği ile bulunup ekleneceği
3. Eklenecek kullanıcının ikinci parametrede verilen özelliğinin değeri

```
PrincipalContext principalContext = new
PrincipalContext(ContextType.Domain, "saitorhan.local",
"Administrator", "123456qaZ.");
GroupPrincipal group =
GroupPrincipal.FindByIdentity(principalContext, "Help Desk");
group.Members.Add(principalContext,
IdentityType.SamAccountName, "saitorhan");
group.Save();
```

## **AD Grup Üyelerinden Kullanıcı Silmek**

Gruptan kullanıcı silinirken de ekleme ile aynı adımlar takip edilir. Aradaki tek fark Add metodu yerine Remove metodu kullanılır.

```
PrincipalContext principalContext = new
PrincipalContext(ContextType.Domain, "saitorhan.local",
"Administrator", "123456qaZ.");
GroupPrincipal group =
GroupPrincipal.FindByIdentity(principalContext, "Help Desk");
group.Members.Remove(principalContext,
IdentityType.SamAccountName, Username);
```

```
group.Save();
```

## AD Kullanıcı Parolası Doğrulama

PrincipalContext.ValidateCredentials metodu, parametre olarak aldığı kullanıcı adı ve parolanın eşleşip eşleşmediğini kontrol eder. İkinci parametrede verilen parola ilk parametrede verilen kullanıcıya aitse "true" yanlışı şifre ise de "false" değer döner.

```
PrincipalContext principalContext = new  
PrincipalContext(ContextType.Domain, "saitorhan.local",  
"Administrator", "123456qaZ.");  
bool credentials =  
principalContext.ValidateCredentials("bilalorhan",  
"Parolam1234");
```

## AD Kullanıcı Şifresi Değiştirme

AD kullanıcılarına ait şifreyi değiştirmek için UserPrincipal sınıfının SetPassword metodu kullanılır.

```
PrincipalContext principalContext = new  
PrincipalContext(ContextType.Domain, "saitorhan.local",  
"Administrator", "123456qaZ.");  
UserPrincipal userPrincipal =  
UserPrincipal.FindByIdentity(principalContext,  
IdentityType.SamAccountName, "bilalorhan");  
userPrincipal.SetPassword("Parolam1234");  
userPrincipal.Save();
```



## AD Kullanıcıyı İlk Oturum Açmada Parola Değiştirmeye Zorlama

Kullanıcıyı ilk oturumda parola değiştirmeye zorlamak için öncelikle "userPrincipal.PasswordNeverExpires = false" ifadesi ile kullanıcının şifresinin süresiz geçerli olması iptal edilir. İkinci adımda "userPrincipal.ExpirePasswordNow()" metodu ile kullanıcının parolasını süresi sona ermişe çekilir.

```
PrincipalContext principalContext = new
PrincipalContext(ContextType.Domain, "saitorhan.local",
"Administrator", "123456qaZ.");
UserPrincipal userPrincipal =
UserPrincipal.FindByIdentity(principalContext,
IdentityType.SamAccountName, userName);
userPrincipal.PasswordNeverExpires = false;
userPrincipal.ExpirePasswordNow();
userPrincipal.Save();
```

Kaynak

Kodlar:

<https://github.com/saitorhan/CSharpForSystemAdmins>