

## Re: VB.NET LDAP Class

**Source:**

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2004-07/0378.html>

---

**From:** Joe Kaplan \((MVP - ADSI)\) (*joseph.e.kaplan\_at\_removethis.accenture.com*)

**Date:** 07/30/04

Date: Fri, 30 Jul 2004 10:29:06 -0500

I wasn't criticizing your code Jon, I was criticizing the code in the article that Raterus pointed to when he suggested that you should have just used it as an example instead. That is a KBase article and needs to be held to a higher standard. It is a big pet peeve of mine.

Your code is basically fine by me! Sorry for the confusion :)

Joe K.

"Jon Delano" <jd31068@hotmail.com> wrote in message news:sgsOc.216053\$XM6.171248@attbi\_s53...

> *Well slap someone for just trying to help out some.*

>

> *I made no comments that it was the best option ... only that it worked for me in what I was doing (I did use AuthenticationTypes.Secure in my code by the way and I have added the dispose and set the objects to nothing.)*

>

> *At the very least it might give someone a starting point if they are lost.*

>

>

>

> *"Joe Kaplan (MVP - ADSI)" <joseph.e.kaplan@removethis.accenture.com> wrote in message news:%23LMMvibdEHA.2384@TK2MSFTNGP09.phx.gbl...*

>> *Actually, the code in that article is pretty bad and contains a number of*

>> *flaws that in my opinion make it not worthy of emulation.*

>>

>> *Problems in the IsAuthenticated method include:*

>> - *They don't use AuthenticationTypes.Secure, so password is sent in clear*

>> *text over the network (!)*

>> - *They don't properly call Dispose on the IDisposable objects, so memory*

>> *leaks will occur, especially given the bugs in the Finalize method for*

>> *DirectoryEntry in .NET 1.0 and 1.1*

>> - *They make a bad assumption about the source the failure by catching the*

> > *general Exception class instead of looking for the "bad credentials"*  
> *HRESULT*  
> > *on the COMException that should be thrown*  
> > *– They make some bad assumptions that won't necessarily work in a*  
> > *multi-domain forest about looking up the user's user name*  
> >  
> > *Problems in the GetGroups methods include:*  
> > *– MemberOf attribute includes non-security groups, does not included*  
> *nested*  
> > *group membership and does not include the primary group, so essentially*  
> *that*  
> > *isn't the correct list*  
> > *– Use the CN attribute for making a security decision instead of*  
> > *sAMAccountName even though CN may not be unique in the directory (only*  
> *the*  
> > *current container)*  
> > *– Parse the DN with a naive check ",", even though DN's can contain ", "*  
> > *escaped with \*  
> > *– Fail to clean up and catch proper exception types as above*  
> >  
> > *As you can see, I'm pretty grumpy about that sample code. Perhaps*  
*someday*  
> > *I'll post something better or it can be fixed. In the mean time, please*  
> > *don't use it.*  
> >  
> > *Joe K.*  
> >  
> >  
> > *"Raterus" <raterus@spam.org> wrote in message*  
> > *news:eRYH1ladEHA.3212@TK2MSFTNGP12.phx.gbl...*  
> > *I hope you didn't spend too long converting this.. :-)*  
> > <http://support.microsoft.com/default.aspx?scid=kb;EN-US;326340>  
> >  
> > *"Jon Delano" <jd31068@hotmail.com> wrote in message*  
> > *news:94cOc.209721\$Oq2.196851@attbi\_s52...*  
> > > *Hello*  
> > >  
> > > *After some effort I was able to come up with this class (converted*  
*from*  
> *a*  
> > *C#*  
> > > *example on MS' web site).*  
> > > *Put this in a class in you vb.net web project and you should be able*  
*to*  
> > > *authenticate a user against active directory and also retrieve their*  
> *user*  
> > > *groups.*  
> > >  
> > > *You must replace the LDAP://yourdomain with the actual domain that*  
*you'd*  
> > > *like to authenticate against.*

```
>>>
>>> Please do with this as you will (and use at your own risk):
>>>
>>> Imports System.DirectoryServices
>>> Imports System.Runtime.InteropServices
>>> Imports System.Globalization
>>>
>>> Public Class ADAAuthenticate
>>> Private _path As String
>>> Private _filterAttribute As String
>>> Private _UserFirstName As String
>>> Private _UserLastName As String
>>> Private _UserPath As String
>>> Private _AuthenticationErrorString As String
>>>
>>> Public Function IsAuthenticated(ByVal UserName As String, ByVal
>> Password
>>> As String) As Boolean
>>> ' authenticate the user and get some info about them
>>> Dim entry As New DirectoryEntry("LDAP://yourdomain", UserName,
>>> Password, System.DirectoryServices.AuthenticationTypes.Secure)
>>> Dim ds As New DirectorySearcher(entry)
>>> Dim myFilter As String =
"&(objectClass=user)(samaccountname="
> +
>>> UserName + ")")
>>>
>>> ds.Filter = myFilter
>>> ds.PropertiesToLoad.Add("sn")
>>> ds.PropertiesToLoad.Add("GivenName")
>>>
>>> Try
>>> Dim sRslt As SearchResult = ds.FindOne
>>> If sRslt Is Nothing Then
>>> Return False
>>> Else
>>> _filterAttribute = UserName
>>> Dim propName As String
>>> Dim value As Object
>>>
>>> For Each propName In sRslt.Properties.PropertyNames
>>> For Each value In sRslt.Properties(propName)
>>> If propName = "sn" Then
>>> _UserLastName = value
>>> End If
>>>
>>> If propName = "givenname" Then
>>> _UserFirstName = value
>>> End If
>>>
>>> If propName = "adspath" Then
```

```
>>> _UserPath = value
>>> End If
>>>
>>> Next value
>>> Next propName
>>>
>>> Return True
>>> End If
>>>
>>> sRslt = Nothing
>>> ds = Nothing
>>>
>>> Catch ex As Exception
>>> _AuthenticationErrorString = ex.Message & "<br>" &
>> ex.StackTrace
>>> Return False
>>> Finally
>>> entry.Dispose()
>>> entry = Nothing
>>> ds.Dispose()
>>> ds = Nothing
>>> End Try
>>>
>>> End Function
>>>
>>> Public ReadOnly Property LdapAuthenticationErrorString() As String
>>> Get
>>> Return _AuthenticationErrorString
>>> End Get
>>> End Property
>>>
>>> Public ReadOnly Property LdapUserFirstName() As String
>>> Get
>>> Return _UserFirstName
>>> End Get
>>> End Property
>>>
>>> Public ReadOnly Property LdapUserLastName() As String
>>> Get
>>> Return _UserLastName
>>> End Get
>>> End Property
>>>
>>> Public Function GetUserGroups(ByVal UserName As String, ByVal
> Password
>>> As String) As String
>>> ' get the groups the user belongs to
>>> Dim entry As New DirectoryEntry("LDAP://yourdomain", UserName,
>>> Password, System.DirectoryServices.AuthenticationTypes.Secure)
>>> Dim search = New DirectorySearcher(entry)
>>>
```

```
>>> search.Filter = "&(objectClass=user)(cn=" + _filterAttribute
+
>> ")"
>>> search.PropertiesToLoad.Add("memberOf")
>>>
>>> Dim groupNames As New System.Text.StringBuilder()
>>>
>>> Try
>>> Dim result As SearchResult = search.FindOne()
>>> Dim propertyCount As Int32 =
>> result.Properties("memberOf").Count
>>> Dim dn As String
>>> Dim equalsIndex As Int32, commaIndex As Int32
>>> Dim propertyCounter As Int32
>>>
>>> For propertyCounter = 0 To propertyCount - 1
>>> dn = result.Properties("memberOf")(propertyCounter)
>>> equalsIndex = dn.IndexOf("=", 1)
>>> commaIndex = dn.IndexOf(",", 1)
>>> If (-1 = equalsIndex) Then
>>> groupNames.Append(dn)
>>> Else
>>> groupNames.Append(dn.Substring((equalsIndex + 1),
>>> (commaIndex - equalsIndex) - 1))
>>> groupNames.Append("/")
>>> End If
>>>
>>> Next propertyCounter
>>>
>>> Catch ex As Exception
>>> Throw New Exception("Error obtaining group names. " +
>>> ex.Message)
>>> Finally
>>> entry.Dispose()
>>> entry = Nothing
>>> search = Nothing
>>> End Try
>>>
>>> Return groupNames.ToString()
>>>
>>> End Function
>>> End Class
>>>
>>> Good luck
>>> Jon
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```