

# System.Byte[] And System.\_\_ComObject in AD LDAP Browser

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2007-11/msg00062.html>

---

- *From:* Darren (UK) <DarrenUK@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
  - *Date:* Thu, 1 Nov 2007 13:33:03 -0700
- 

Hello,

I have a small issue with some of the information returned in an LDAP browser. On certain attributes it returns either System.Byte[] And System.\_\_ComObject

I suspect this is a result of the data held. For example i know that PwdLastSet attrib attrib is stored as Int64 values. How do i determine what System.Byte[] And System.\_\_ComObject are??

A little research has yielded the following:

If you use the DirectorySearcher, it marshals AD INTEGER8 types to .NET Int64 automatically, so no work needs to be done. This is by far the easiest way.

If you use the DirectoryEntry, it marshals the value as an ADSI IADsLargeInteger type, which at runtime is a System.\_\_ComObject. This is annoying, but you can get the value with a little interop and some data munging. This works for me:

```
dim entry as new DirectoryEntry("LDAP://yourdn here")
dim pwd as object = entry.Properties("pwdLastSet").Value
dim pwdDate as DateTime
pwdDate = DateTime.FromFileTimeUtc(GetInt64FromLargeInteger( pwd))
```

Function GetInt64FromLargeInteger(byval largeInteger as Object) as Int64

```
dim low as int32
dim high as int32
dim valBytes(7) as byte
```

```
dim longInt as IADsLargeInteger = CType(largeInteger, IADsLargeInteger)
low = longInt.LowPart
high = longInt.HighPart
```

```
BitConverter.GetBytes(low).CopyTo(valBytes, 0)
```

## System.Byte[] And System.\_\_ComObject in AD LDAP Browser

```
BitConverter.GetBytes(high).CopyTo(valBytes, 4)
```

```
Return BitConverter.ToInt64(valBytes, 0)
```

```
End Function
```

```
<ComImport(), Guid("9068270b-0939-11D1-8be1-00c04fd8d503"),  
InterfaceTypeAttribute(ComInterfaceType.InterfaceI sDual)> _  
public interface IADsLargeInteger  
property HighPart as int32  
property LowPart as int32  
end interface
```

Note that you can also get the high and low values via reflection and late binding or by importing the activeds.dll COM object into .NET.

Qn: I am new to Vb.net and still learning. How can i incorporate the above to return what i need????

MANY MANY THANKS

----- MY CODE  
-----

```
Imports System  
Imports System.DirectoryServices  
Imports System.DirectoryServices.ActiveDirectory  
Imports System.Windows.Forms
```

```
Public Class LDAPBrowser
```

```
Private Base As DirectoryEntry  
Private str As String()
```

```
Private Sub LDAPBrowser_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
Connect()
```

```
End Sub
```

```
Private Sub tvwADStructure_AfterSelect(ByVal sender As System.Object,  
ByVal e As System.Windows.Forms.TreeViewEventArgs) Handles  
tvwADStructure.AfterSelect
```

```
'Fill the TreeView dynamic after Click  
If e.Node.Nodes.Count = 0 Then
```

## System.Byte[] And System.\_\_ComObject in AD LDAP Browser

```
Dim parent As DirectoryEntry = DirectCast(e.Node.Tag,
DirectoryEntry)

If parent IsNot Nothing Then

If parent.Children IsNot Nothing Then

For Each Iter As DirectoryEntry In parent.Children

Dim childNode As TreeNode =
e.Node.Nodes.Add(Iter.Name)

childNode.Tag = Iter

Next
End If
End If
End If

'Fill the ListView Element
Try

Dim list As DirectoryEntry = DirectCast(e.Node.Tag,
DirectoryEntry)

If list IsNot Nothing Then

lstAttribs.Clear()

'Add some information to ListView Element
lstAttribs.Columns.Add("Attribute", 90,
HorizontalAlignment.Left)
lstAttribs.Columns.Add("Value", 350, HorizontalAlignment.Left)

For Each listIter As Object In list.Properties.PropertyNames

For Each Iter As Object In
list.Properties(listIter.ToString())
Dim item As New
System.Windows.Forms.ListViewItem(listIter.ToString(), 0)

item.SubItems.Add(Iter.ToString())

lstAttribs.Items.AddRange(New ListViewItem() {item})

Next
Next
End If

Catch ex As System.Exception
```

## System.Byte[] And System.\_\_ComObject in AD LDAP Browser

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub Connect()

'Pass Connect info to DirectoryEntry object:

Base = New DirectoryEntry("LDAP://XX.XXXXX.com)

'Read the root:

If Base IsNot Nothing Then

tvwADStructure.Nodes.Clear()

tvwADStructure.BeginUpdate()

Dim childNode As TreeNode = tvwADStructure.Nodes.Add(Base.Name)

childNode.Tag = Base

Try

For Each rootIter As DirectoryEntry In Base.Children

Dim RootNode As TreeNode =

childNode.Nodes.Add(rootIter.Name)

RootNode.Tag = rootIter

Next

Finally

childNode.Expand()

tvwADStructure.EndUpdate()

End Try

End If

End Sub

End Class

⋮