

Forum:[Hack Forums](#)

Thread: [Serialization and Deserialization](#)

Vysse - June 20, 2015, 7:35 pm

These quick snippets allow you to turn any object into a byte array and then back into said object. Perfect for RATs.

```
Code Public Shared Function Serialize(ByVal obj As Object) As Byte()  
    Using ms As New MemoryStream  
        Dim bf As New BinaryFormatter  
        bf.Serialize(ms, obj)  
        Return ms.ToArray  
    End Using  
End Function  
  
Public Shared Function DeSerialize(Of T)(ByVal data As Byte()) As T  
    Using ms As New MemoryStream(data)  
        Dim bf As New BinaryFormatter  
        Return DirectCast(bf.Deserialize(ms), T)  
    End Using  
End Function
```

Here is a quick example of how to use it.

```
Code Dim img As Image = retFlag("ipaddress") 'retFlag is an image function I am using in a project  
of mine  
Dim imgByte() As Byte = Serialize(img)    Public Shared Function Serialize(ByVal obj As  
Object) As Byte()  
    Dim img2 As Image = DeSerialize(Of Image)(imgByte)  
    PictureBox1.Image = img2
```

Corneliuss - June 21, 2015, 2:15 pm

You should dispose the MemoryStream or use the "Using statement" which automatically disposes it

Vysse - June 21, 2015, 3:13 pm

(Jun 21, 2015 02:15 PM)Corneliuss Wrote: [\[CITE\]](#) You should dispose the MemoryStream or use the "Using statement" which automatically disposes it

I am not sure how that would work since the code exits the function once it reaches the return statement doesn't it?

Odus - June 21, 2015, 3:35 pm

(Jun 21, 2015 03:13 PM)Vysse Wrote: [\[CITE\]](#) I am not sure how that would work since the code exits the function once it reaches the return statement doesn't it?

Nope. afaik, the Using block guarantees object disposal since it's already marked for collection by the time the function is exited.

Vysse - June 21, 2015, 3:44 pm

(Jun 21, 2015 03:35 PM)Odus Wrote: [\[CITE\]](#) Nope. afaik, the Using block guarantees object disposal since it's already marked for collection by the time the function is exited.

Fixed and updated. Thanks for the input.

