

Forum: [Hack Forums](#)

Thread: [\[C#\] Getting the HEX](#)

codepoet - May 1, 2010, 4:26 pm

Used to convert a byte array to its hex equivalent.

```
Code    public static string ByteToHexString(byte[] p)
    {
        char[] c = new char[p.Length * 2 + 2];

        byte b;

        c[0] = '0'; c[1] = 'x';

        for (int y = 0, x = 2; y < p.Length; ++y, ++x)
        {
            b = ((byte)(p[y] >> 4));

            c[x] = (char)(b > 9 ? b + 0x37 : b + 0x30);

            b = ((byte)(p[y] & 0xF));

            c[++x] = (char)(b > 9 ? b + 0x37 : b + 0x30);
        }

        return new string(c);
    }
```

originally written by: RuE

If You Like this tutorial,
please check out my other tutorials by clicking ==> [here](#)

julian - May 1, 2010, 5:12 pm

Or maybe this:

```
Code public static string ByteToHexString(byte[] bytes)
{
    return String.Concat((from b in bytes select b.ToString("x2")).ToArray<String>());
}
Code Public Shared Function ByteToHexString(ByVal bytes() As Byte) As String
    Return String.Concat((From b In bytes Select b.ToString("x2")).ToArray) ' X2 for upper case hex
End Function
```

Linq is kinda ugly and confusing though IMHO. Or this:

```
Code public static string ByteToHexString(byte[] bytes)
{
    return BitConverter.ToString(bytes);
}
Code Public Function ByteToHexString(ByVal bytes() As Byte) As String
    Return BitConverter.ToString(bytes)
```

End Function This one is so short there's no point putting it in a separate function really. It generates Strings that look like "00-01-AA-FF". If you want it to look different then you'll have to do it manually.

codepoet - May 1, 2010, 5:39 pm
thanks for translating

julian - May 1, 2010, 5:45 pm
Argh - Interlocked.Increment...

Here's another, in VB.Net:

```
Code Public Function ByteToHexString(ByVal bytes() As Byte, Optional ByVal uppercase As Boolean = False, Optional ByVal separator As String = Nothing) As String
    Dim formatString As String = If(uppercase, "X2", "x2")
    Dim sb As New System.Text.StringBuilder(bytes.Length * 2)
    For i As Integer = 0 To bytes.Length - 1
        sb.Append(bytes(i).ToString(formatString))
        If separator IsNot Nothing Then
            sb.Append(separator)
        End If
    Next
    If separator IsNot Nothing Then
        sb.Remove(sb.Length - separator.Length, separator.Length)
    End If
    Return sb.ToString
End Function
```

This one lets you say if you want caps, and if you want some separator (argh, I spelled separator wrong) string to appear between the hex digits. For example:

```
Code Dim rand As New Random
Dim bytes(5) As Byte
rand.NextBytes(bytes)
Console.WriteLine(ByteToHexString(bytes))
Console.WriteLine(ByteToHexString(bytes, True))
Console.WriteLine(ByteToHexString(bytes, False, ", "))
Console.WriteLine(ByteToHexString(bytes, False, "-"))
output:
6bb0727eae52
6BB0727EAE52
6b, b0, 72, 7e, ae, 52
6b-b0-72-7e-ae-52
```

codepoet - May 1, 2010, 5:47 pm
thanks for the new function