

microsoft.public.dotnet.languages.csharp: Re: Writing extended ascii characters to text file.

Re: Writing extended ascii characters to text file.

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-01/4122.html>

From: Dmitriy Lapshin [C# / .NET MVP] (*x-code_at_no-spam-please.hotpop.com*)

Date: 01/19/05

Date: Wed, 19 Jan 2005 14:12:56 +0200

Hi JSM,

First of all, remember that all strings in .NET are internally Unicode strings, so in order to get real ASCII codes you should use the GetBytes method of an Encoding instance configured for the ASCII encoding (as far as I remember there is a static instance accessible as Encoding.Ascii). Now that you have the bytes with the ASCII codes, you do the encryption and again, you've got bytes, not characters. Therefore, it seems natural to write out the resultant bytes in a binary mode where the issue of character encoding simply is out of the picture.

--

Sincerely,

Dmitriy Lapshin [C# / .NET MVP]

Bring the power of unit testing to the VS .NET IDE today!

<http://www.x-unity.net/teststudio.aspx>

"JSM" <jsm@spam.pacific.net.au> wrote in message

news:eVpZJRh\$EHA.1400@TK2MSFTNGP11.phx.gbl...

> Hi,

>

> I am just trying to port an existing simple encryption routine to C#. this

> routine simply adds/subtracts 10 ascii characters to each character in a

> text file (except quotes). The routine for decrypting the file works fine

> however when I encrypt the file, several characters are corrupted. when I

> looked into it they are always extended ascii characters (eg "x" which is

> ascii character 120 gets translated to ascii character 130 which is part

> of

> the extended ascii range of characters). I am assuming this has something

> to

> do with the Encoding which I used to read/write the text files but I just

> can't seem to get it to work. I have tried specifying all different types

> of

> encoding (Unicode, Ascii, UTF7/8 without success).

>

> I can't change the encryption method because this needs to work with

> existing installations of my application.

>

> Below are the two routines. Any ideas ? I thought this would be a very

> simple task but alas I was wrong!

>

> Cheers,

>

> John

Re: Writing extended ascii characters to text file.

microsoft.public.dotnet.languages.csharp: Re: Writing extended ascii characters to text file.

```
> -----
>
> public void DecryptFile(string SourceFile,string DestFile)
> {
>     string line;
>     StreamReader sr=new StreamReader(SourceFile,Encoding.Default);
>     StreamWriter sw=new StreamWriter(DestFile,false,Encoding.Default);
>
>     while ((line=sr.ReadLine())!=null)
>     {
>         string newline="";
>
>         for (int x=0;x<line.Length;x++)
>         {
>             if ((line.Substring(x,1)==qte) ||
> (line.Substring(x,1)==Convert.ToChar(44).ToString()))
>                 newline +=line.Substring(x,1);
>             else
>             {
>                 newline +=
> Convert.ToChar((int)Encoding.Default.GetBytes(line.Substring(x,1))[0]-10).To
> String();
>             }
>         }
>         sw.WriteLine(newline);
>     }
>
>     sr.Close();
>     sw.Close();
> }
>
>
> public void EncryptFile(string SourceFile,string DestFile)
> {
>     string line;
>     StreamReader sr=new StreamReader(SourceFile,Encoding.Default);
>     StreamWriter sw=new StreamWriter(DestFile,false, Encoding.Default);
>
>     while ((line=sr.ReadLine())!=null)
>     {
>         string newline="";
>         for (int x=0;x<line.Length;x++)
>         {
>             if ((line.Substring(x,1)==qte) ||
> (line.Substring(x,1)==Convert.ToChar(44).ToString()))
>                 newline +=line.Substring(x,1);
>             else
>             {
>                 newline +=
> Convert.ToChar((int)Encoding.Default.GetBytes(line.Substring(x,1))[0]+10).To
> String();
>             }
>         }
>
>         sw.WriteLine(newline);
>     }
>
>     sr.Close();
>     sw.Close();
> }
>
>
```