

Forum: [Hack Forums](#)

Thread: [\[C#\]\[SRC\]ROT13 Class](#)

Ethereal - January 11, 2011, 9:32 am

This is my own ROT13 class. It is fast - it did 10,000 characters instantly(1/5 of a second) and I am pretty proud of it.

Some constructive criticism is Ok, but don't flame.

Code class ROT13

```
{
    /// <summary>
    /// This is an implementation of the ROT13(Rotation of 13, a.k.a Caesar Shift) algorithm, coded
    by Ethereal of HackForums.net.
    /// This will scramble your string into an unreadable form.
    /// </summary>
    public string AlterText(string input)
    {
        char[] text = input.ToCharArray();
        for (int i = 0; i < input.Length; i++)
        {
            int value = (int)text[i];
            if (value >= 'a' && value <= 'm' || value >= 'A' && value <= 'M')
            {
                value += 13;
            }
            else if (value >= 'm' && value <= 'z' || value >= 'M' && value <= 'Z')
            {
                value -= 13;
            }

            text[i] = Convert.ToChar(value);
        }
        string FinVal = new string(text);
        return FinVal;
    }
}
```

I hope that it is useful to you.

Update* Vigenere and ROT_X class to come soon. One more bug fix for Vigenere.

Blue Screen of Death - January 11, 2011, 9:33 am

why don't you make a ROT-X class? not just a ROT-13 ...

Anyway i like it.

Thanks

haphap - January 11, 2011, 9:35 am

you don't have to get the int value of the char... you just do char > 'a' etc it auto converts

Ethereal - January 11, 2011, 9:37 am

(Jan 11, 2011 10:33 AM)bomblader Wrote: [\[CITE\]](#) why don't you make a ROT-X class? not just a ROT-13 ...

Anyway i like it.

Thanks

I will be updating it to such a class. I was trying to keep it simple for my first implementation.

(Jan 11, 2011 10:35 AM)haphap Wrote: [\[CITE\]](#) you don't have to get the int value of the char... you just do char > 'a' etc it auto converts

Thank you for that. This is the kind of info that I am looking for!

BravoTwoZero - January 11, 2011, 10:43 am

Very nice... will try and test it but first I need to learn C# [:(]

Ethereal - January 11, 2011, 12:23 pm

(Jan 11, 2011 11:43 AM).Warlock. Wrote: [\[CITE\]](#) Very nice... will try and test it but first I need to learn C# [:(]

I would definetly recommend it.

I may translate this to VB, when I finish a ROT - X class.

BravoTwoZero - January 11, 2011, 12:55 pm

(Jan 11, 2011 01:23 PM)Ethereal Wrote: [\[CITE\]](#) I would definetly recommend it.

I may translate this to VB, when I finish a ROT - X class.Ok thankyou , looking forward to next three months which I havw the GSCE's coming up then will start learning C sharp... Thank god!!

Mr. Anderson - January 11, 2011, 1:41 pm

Ha nice! Glad people are starting to port encryption to .net.

AssChin79 - January 14, 2011, 7:02 am

Nice Caesar Cipher! How 'bout a Vigenere cipher next! That one's my favorite for obscuring symmetric keys!

Ethereal - January 15, 2011, 3:58 am

(Jan 14, 2011 08:02 AM)AssChin79 Wrote: [\[CITE\]](#) Nice Caesar Cipher! How 'bout a Vigenere cipher next! That one's my favorite for obscuring symmetric keys!

I've been working on one.

It's all working fine, no errors, except when I decrypt, it changes the encrypted 'z' into an 'a'.

Breakpoints time.

Zer0Fact0r - January 15, 2011, 6:38 am

Look into this: [http://msdn.microsoft.com/en-US/library/6373h346\(v=VS.80\).aspx](http://msdn.microsoft.com/en-US/library/6373h346(v=VS.80).aspx)

Also this is not a proper Caesar cipher. A Caesar cipher with a shift of 1 will translate a 'z' to a 'a'.

Ethereal - January 15, 2011, 7:26 am

(Jan 15, 2011 07:38 AM)Zer0Fact0r Wrote: [\[CITE\]](#) Look into this:
[http://msdn.microsoft.com/en-US/library/6373h346\(v=VS.80\).aspx](http://msdn.microsoft.com/en-US/library/6373h346(v=VS.80).aspx)

Also this is not a proper Caesar cipher. A Caesar cipher with a shift of 1 will translate a 'z' to a 'a'. This is a Caesar Cipher of 13. Trust me, I know what I am talking about.

This is ROT13, not generic Caesar Cipher.

Also, I have since updated the class with the '||' operator, so I will update it soon.