

Forum:[Hack Forums](#)

Thread: [Converting Base 02 to Base 10 \[help\]](#)

SIM-1327690 - December 19, 2012, 8:41 pm

Was wondering if anyone was able to help me out on this.
I need a function to convert base 02 (binary) to base 10.
Thanks everyone for your time!

TrackPro+ - December 19, 2012, 8:47 pm

Have you tried here first? <http://goo.gl/L1zZw>

I don't know anything about this, not trying to be sarcastic or anything, but yeah. Just checking

li0nsar3c00l - December 19, 2012, 10:39 pm

[http://msdn.microsoft.com/de-de/library/aa326741\(v=vs.71\).aspx](http://msdn.microsoft.com/de-de/library/aa326741(v=vs.71).aspx)

Did you checked out this yet?

0xDEAD10CC - December 20, 2012, 12:30 am

This is the difference between Base2 and Base10. But what you've stated in your post doesn't conform with what the title says, so which is it that you are trying to do?

(Dec 19, 2012 09:41 PM)Mijoverse Wrote: [\[CITE\]](#) Was wondering if anyone was able to help me out on this.

I'm writing a multi-layered encryption algorithm and have fine tuned it to the point I just need to be able to create a function that converts an 8-bit byte into its decimal form.

Thanks everyone for your time!

A) Convert each byte (8-bit) into Decimal - Byte > Decimal.

B) Convert the 8 bits to its Decimal representative.

???

?

```
Code Dim b As Byte = &HFF
Dim i As Integer = CInt(b)
Console.WriteLine(i)
Console.ReadKey()
Code '255 (Dec) = 11 (Binary)
Dim bin As Integer = Convert.ToInt32("11", 2)
Console.WriteLine(bin)
Console.ReadKey()
```

SIM-1327690 - December 20, 2012, 5:57 pm

I guess I should've just said converting from base-2 to base 10, sorry for that confusion.
Thanks everyone for your input!

+Bourbon - December 20, 2012, 8:04 pm

Here are two algorithms I wrote earlier in the year that you can use. You only asked for the first but you (or anybody else) could make use of the second in the future :p

0xDEAD10CC - December 20, 2012, 8:12 pm

Perhaps a better structure for functions would be to make them conform to the TryParse() paradigm. I don't see much benefit in making it return an error string when that same returned string value could also be valid output. It just means you'd have to check the output yourself after the fact anyways, which is mainly a waste of time if you already knew at one point that it was invalid.

It's like having one person go to make sure that he took care of something, and then because this person doesn't tell anybody else anything for confirmation, another person always has to come and check after the fact anyways. Not very efficient.

Returning error strings and mixing the return value with good output is bad habit, and also defined in coding practices provided by MSDN.

+Bourbon - December 20, 2012, 9:07 pm

(Dec 20, 2012 09:12 PM)ArkPhaze Wrote: [\[CITE\]](#) Perhaps a better structure for functions would be to make them conform to the TryParse() paradigm. I don't see much benefit in making it return an error string when that same returned string value could also be valid output. It just means you'd have to check the output yourself after the fact anyways, which is mainly a waste of time if you already knew at one point that it was invalid.

It's like having one person go to make sure that he took care of something, and then because this person doesn't tell anybody else anything for confirmation, another person always has to come and check after the fact anyways. Not very efficient.

Returning error strings and mixing the return value with good output is bad habit, and also defined in coding practices provided by MSDN.

You did not see the context in which I implemented these functions. I used these functions to return a value to print to a console window. In procedural programming, functions are used to break down code. Rather than creating a long method with ambiguous self-documentation I broke the control in to separate functions which is perfectly acceptable. There is no such thing as a be all end and all design pattern, you know.

0xDEAD10CC - December 20, 2012, 11:16 pm

(Dec 20, 2012 10:07 PM)ByteBlast Wrote: [\[CITE\]](#) You did not see the context in which I implemented these functions. I used these functions to return a value to print to a console window. In procedural programming, functions are used to break down code. Rather than creating a long method with ambiguous self-documentation I broke the control in to separate functions which is perfectly acceptable. There is no such thing as a be all end and all design pattern, you know.

No, but there are standards, like OOP. You don't have to do everything OOP, but it's probably better that you do as it upholds a good design pattern. You don't have to follow the strict guidelines given by the nature of OOP, but it's also still better that you probably do so.

To the point: The reason being is because, now if he goes to use your function, or anybody else for that matter, what if they do not use it to write to the console in the way you do, and they need the validation? Not saying they can't modify your function, but that's why I will point this out. Because I know that there's probably some out there who will take your function for granted, and do something stupid like this:

```
Code If ByteBlastsFunction() = "This is the error value from the function" Then
'Error
Else
'Get to it
End If
```

SIM-1327690 - December 23, 2012, 5:21 pm

Yeah guys so just looking for a function to convert base 02 to base 10, if anyone can provide help it'd be much appreciated.

+Bourbon - December 23, 2012, 6:37 pm

There you go, finally!

infamousnoone - December 23, 2012, 6:54 pm

.....

sorry4this - December 23, 2012, 9:47 pm

Hey HF

Im also have one question about convertation text to hex and revert

Im using this class ([pastebin](#)) and its working only with text. When im trying to convert any other symbols, like vbNewline, vbTab etc. im got stuck of output (screenshot)

Spoiler ([Click to View](#))[img]http://imageplay.net/img/tya22299119/01.jpg[/img]Can anyone help me?