

## Re: Price Data Formatting

---

*Source:*

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2008-08/msg00742.html>

---

- *From:* Mike Williams <[gagamomo@xxxxxxxxxxx](mailto:gagamomo@xxxxxxxxxxx)>
  - *Date:* Mon, 18 Aug 2008 21:08:01 -0700 (PDT)
- 

On 18 Aug, 22:37, "Webbiz" <[nore...@xxxxxxx](mailto:nore...@xxxxxxx)> wrote:

What I have been doing now is loading the data and noting the maximum digits found right of the decimal and left of the decimal. How I do this is through a LOOP, using INSTR

I can't seem to get answers through the MS server at the moment so I'm posting this from a Google Groups account. Anyone else having this problem at the moment? Anyway, you don't need to do any of that stuff. The Format function will do it all for you. Try this:

```
Private Sub Command1_Click()  
Dim j As Single  
j = 12.34  
Print Format(j, "0.0000")  
j = 0.345  
Print Format(j, "0.0000")  
End Sub
```

In the line `Format(j, "0.0000")` the first zero tells VB that you want it to insert a leading zero before the decimal point if the number does not have a "whole" part. Otherwise, if you did not want the leading zero you would instead use `Format(j, "#.0000")`. The second four zeros simply tell VB how many decimal digits you want to display (four in this case) and VB will then add sufficient zeros to the output to make it have the correct number of decimal places.

In this case the example is "hard coded" to format to four decimal places, specified by the four zeros after the decimal point in the line `at(j, "#.0000")`. However, if you do not know at design time how many decimal places you require you can "build" the string `"#.0000"` at runtime, inserting the appropriate number of zeros into the Format instruction, so that you could for example build the string `"#.000"` if your code decided at runtime that it wanted three decimal places. If that's what you need to do and if you want any help with it then post again.

## Re: Price Data Formatting

When you print out the results of course, as the above simple example does, the output is printed left aligned. If you instead want it to be right aligned (as is often the case with such numbers) then you'll need to add more code, but I haven't included any such code in this example because much depends on how you are outputting the result (printing it to a picturebox or a printer or something or perhaps displaying it in some kind of control such as a ListBox or a Label or whatever)

Mike