

If, IIf() and If()

Asked 16 years, 3 months ago Modified 10 years, 7 months ago

Viewed 5k times



12

I recently asked a question about [IIf vs. If](#) and found out that there is another function in VB called **If** which basically does the same thing as **IIf** but is a short-circuit.



Does this **If** function perform better than the **IIf** function?
Does the **If** statement trump the **If** and **IIf** functions?



.net

vb.net

if-statement

iif-function

Share

Improve this question

Follow

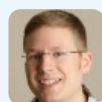
edited May 23, 2017 at 11:52



Community Bot

1 • 1

asked Aug 26, 2008 at 16:03



Bryan Roth

10.7k • 15 • 49 • 56

2 Answers

Sorted by:

Highest score (default)



15

Damn, I really thought you were talking about the operator all along. ;-) Anyway ...



Does this If function perform better than the IIf function?

Definitely. Remember, it's built into the language. Only one of the two conditional arguments has to be evaluated, potentially saving a costly operation.

Does the If statement trump the If and IIf functions?

I think you can't compare the two because they do different things. If your code semantically performs an assignment you should emphasize this, instead of the decision-making. Use the `If` operator here instead of the statement. This is especially true if you can use it in the initialization of a variable because otherwise the variable will be default initialized, resulting in slower code:

```
Dim result = If(a > 0, Math.Sqrt(a), -1.0)
```

' versus

```
Dim result As Double ' Redundant default initialization
If a > 0 Then
    result = Math.Sqrt(a)
Else
    result = -1
End If
```

Share Improve this answer

Follow

answered Aug 26, 2008 at 16:10



Konrad Rudolph

545k ● 138 ● 956 ● 1.2k



3



One very important distinct between `IIf()` and `If()` is that with `Option Infer On` the later will implicitly cast the results to the same data type in certain cases, as where `IIf` will return `Object`.

Example:

```
Dim val As Integer = -1
Dim iifVal As Object, ifVal As Object
iifVal = IIf(val >= 0, val, Nothing)
ifVal = If(val >= 0, val, Nothing)
```

Output:

`iifVal` has value of `Nothing` and type of `Object`

`ifVal` has value of `0` and type of `Integer`, b/c it is implicitly converting `Nothing` to an `Integer`.

Share Improve this answer

answered Apr 30, 2014 at 18:45

Follow



BateTech

6,436 ● 3 ● 22 ● 33

- 1 Happened last month in my code and I spent an hour debugging to figure this out. This is a drawback of `If()` – RSK
Dec 9, 2020 at 3:26

