

Forum:[Hack Forums](#)

Thread: [\[Preview\] CPU and Ram Usage Meter](#)

Ebony - January 27, 2014, 8:26 pm

Hello everyone on HF. Im just showing off my little CPU and RAM meter that I made a few days ago I got a new computer, and since my old computer had the widgets, I enjoyed using the cpu and ram meter on that. The computer was windows 7 by the way. And my new Windows 8 computer doesn't have that. Microsoft has canceled all widgets because of a security vulnerability with them. So I decided to make my own. Its simple.

Here is a picture of the Meter >

Spoiler [\(Click to View\)](#)

It also has a little menu from where you can exit or lock the widget to the screen >

Spoiler [\(Click to View\)](#)

Thanks for taking a look and if anyone would like it, I will post a download link along with a virus scan

bonsai - January 27, 2014, 9:13 pm

Post the source, would be much more safe :)!

ecaep42 - January 27, 2014, 10:49 pm

(Jan 27, 2014 10:13 PM)lemayojo Wrote: [\[CITE\]](#) Post the source, would be much more safe :)!

I have this exact same thing on one of my programs I released here, the source is simply using a few performance counters to monitor each core separately.

[img]http://img15.imageshack.us/img15/1769/3l5l.png[/img]

I added 4 cores and 1 ram monitor, so you will need 5 performance counters.

Firstly, declare new integers via coding

Code Dim ram as Integer

Dim core1 as Integer

Dim core2 as Integer

Dim core3 as Integer

Dim core4 as Integer

Now, create your first timer

Open timer1, and paste in this

Code On Error Resume Next

core1 = PerformanceCounter1.NextValue

core2 = PerformanceCounter2.NextValue

ram = PerformanceCounter3.NextValue

core3 = PerformanceCounter4.NextValue

core4 = PerformanceCounter5.NextValueI added OERN because some people don't have 4 cores.

We're gonna start ram first, because CPU can be repeated.

For RAM

Spoiler [\(Click to View\)](#)

Add a Performance County

Change the Category name to Memory.
Change CounterName to % Committed Bytes in Use

```
Timer2 Code If EProgBar3.Value < ram Then
    EProgBar3.Value += 1
    Elself EProgBar3.Value > ram Then
    EProgBar3.Value -= 1
End If
```

Change EProgBar3 to any progressbar you have, or a label, then test it and you're golden.

For the CPU

Spoiler ([Click to View](#))

Add 4 more Performance Counters, name them or don't, i don't really care.

Change the Category name to Processor, and change the counterName to % Processor Time on all 4 counters.

Next, select the first counter you want for the first core, in my case its performance counter 1. Type 0 in the instance name.

Performance counter 2 is core 2, so type 1 in the instance name.

Performance counter 4 is core 3, so type 2 in the instance name.

Performance counter 5 is core 4, so type 3 in the instance name.

Now, open timer 2, and add this code.

```
Code If EProgBar1.Value < core1 Then
    EProgBar1.Value += 1
    Elself EProgBar1.Value > core2 Then
    EProgBar1.Value -= 1
End If
```

```
If EProgBar2.Value < core2 Then
    EProgBar2.Value += 1
    Elself EProgBar2.Value > core2 Then
    EProgBar2.Value -= 1
End If
```

```
If EProgBar4.Value < core3 Then
    EProgBar4.Value += 1
    Elself EProgBar4.Value > core3 Then
    EProgBar4.Value -= 1
End If
```

```
If EProgBar5.Value < core4 Then
    EProgBar5.Value += 1
    Elself EProgBar5.Value > core4 Then
    EProgBar5.Value -= 1
End If
```

You should be good.

Performance counters are great for system monitoring.

Edit: I notice that my source is more in-depth with separate core monitoring, however his should be

just as mine.

Ebony - January 28, 2014, 1:24 am

(Jan 27, 2014 11:49 PM)ecaep42 Wrote: [\[CITE\]](#) I have this exact same thing on one of my programs I released here, the source is simply using a few performance counters to monitor each core separately.

[img]http://img15.imageshack.us/img15/1769/3l5l.png[/img]

I added 4 cores and 1 ram monitor, so you will need 5 performance counters.

Firstly, declare new integers via coding

Code Dim ram as Integer

Dim core1 as Integer

Dim core2 as Integer

Dim core3 as Integer

Dim core4 as Integer

Now, create your first timer

Open timer1, and paste in this

Code On Error Resume Next

core1 = PerformanceCounter1.NextValue

core2 = PerformanceCounter2.NextValue

ram = PerformanceCounter3.NextValue

core3 = PerformanceCounter4.NextValue

core4 = PerformanceCounter5.NextValue I added OERN because some people don't have 4 cores.

We're gonna start ram first, because CPU can be repeated.

For RAM

Spoiler [\(Click to View\)](#)

Add a Performance County

Change the Category name to Memory.

Change CounterName to % Committed Bytes in Use

Timer2 Code If EProgBar3.Value < ram Then

EProgBar3.Value += 1

Elseif EProgBar3.Value > ram Then

EProgBar3.Value -= 1

End If

Change EProgBar3 to any progressbar you have, or a label, then test it and you're golden.

For the CPU

Spoiler [\(Click to View\)](#)

Add 4 more Performance Counters, name them or don't, i don't really care.

Change the Category name to Processor, and change the counterName to % Processor Time on all 4 counters.

Next, select the first counter you want for the first core, in my case its performance counter 1. Type 0 in the instance name.

Performance counter 2 is core 2, so type 1 in the instance name.

Performance counter 4 is core 3, so type 2 in the instance name.

Performance counter 5 is core 4, so type 3 in the instance name.

Now, open timer 2, and add this code.

```
Code If EProgBar1.Value < core1 Then
    EProgBar1.Value += 1
    ElseIf EProgBar1.Value > core2 Then
    EProgBar1.Value -= 1
End If
```

```
If EProgBar2.Value < core2 Then
EProgBar2.Value += 1
ElseIf EProgBar2.Value > core2 Then
EProgBar2.Value -= 1
End If
```

```
If EProgBar4.Value < core3 Then
EProgBar4.Value += 1
ElseIf EProgBar4.Value > core3 Then
EProgBar4.Value -= 1
End If
```

```
If EProgBar5.Value < core4 Then
EProgBar5.Value += 1
ElseIf EProgBar5.Value > core4 Then
EProgBar5.Value -= 1
End If
```

You should be good.

Performance counters are great for system monitoring.

Edit: I notice that my source is more in-depth with separate core monitoring, however his should be just as mine. Wow that looks really amazing. Im impressed. But yes mine is pretty much the same just not as much since its only monitoring one cpu

Nikey646 - January 28, 2014, 1:39 am

Both examples lack Dynamic Multi-CPU support ;((If that even makes sense, basically generating the performance counters based on how many CPU's can be counted :>)

I'll grab the source for the Dynamic CPU Monitor i wrote a while back, and edit my post w/ it's source. It could most likely be improved upon :P

```
Code Dim performance As New List(Of PerformanceCounter)
Dim CPUload As New List(Of Label)
Dim ProgressBar As New List(Of ProgressBar)
Dim locationx As Integer = 12
Dim locationy As Integer = 9
```

```
Dim WithEvents Timer As New Timer With {.Interval = 500, .Enabled = True}
```

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
Me.Size = New Size(300, 95)
```

```
For i As Integer = 0 To System.Environment.ProcessorCount - 1
```

```
Dim processor As New PerformanceCounter  
processor.CategoryName = "Processor"  
processor.CounterName = "% Processor Time"  
processor.InstanceName = i
```

```
performance.Add(processor)
```

```
Dim CPU As New Label  
CPU.Text = "Logical Processor #" + (i + 1).ToString  
CPU.AutoSize = True  
CPU.Location = New Point(locationx, locationy)
```

```
Dim Progress As New ProgressBar  
Progress.Location = New Point(locationx, locationy + 16)  
Progress.Width = 221
```

```
ProgressBar.Add(Progress)
```

```
Dim Load As New Label  
Load.Text = "0%"  
Load.AutoSize = False  
Load.Size = New Size(33, 13)  
Load.Location = New Point(locationx + 227, locationy + 21)
```

```
CPUAdd.Add(Load)
```

```
locationy += 51  
Me.Height += 45
```

```
Me.Controls.AddRange({CPU, Progress, Load})
```

```
Next  
End Sub
```

```
Private Sub TimerTick(ByVal sender As Object, ByVal e As EventArgs) Handles Timer.Tick  
For i As Integer = 0 To performance.Count - 1  
If ProgressBar(i).Value < performance(i).NextValue Then  
ProgressBar(i).Value += 1  
Else  
Try  
ProgressBar(i).Value -= 1  
Catch ex As Exception : End Try  
End If  
CPUAdd(i).Text = ProgressBar(i).Value.ToString + "%"  
Next  
End Sub
```

That'll work if you through it into a Windows Form Application....Most likely.

It could be improved via fixing the Form Sizing (It's incredibly rough, i didn't do the math correctly, so unless the computer has 8 logical processors like mine, it won't look exactly right...), and instead

of using 3 separate Lists of controls, possibly creating a single list of a custom class, which holds all the information related to the performance counter.

There also seems to be a problem with large CPU Activity Spikes (For example, i was going to provide a screenshot while i was encoding a video to make my CPU Activity spike, however it was essentially all 8 logical processors at 99% or 100%...), taking too long to update

ecaep42 - January 28, 2014, 11:03 am

(Jan 28, 2014 02:39 AM)Nikey646 Wrote: [\[CITE\]](#) Both examples lack Dynamic Multi-CPU support ;((If that even makes sense, basically generating the performance counters based on how many CPU's can be counted :>)

I'll grab the source for the Dynamic CPU Monitor i wrote a while back, and edit my post w/ it's source. It could most likely be improved upon :P

```
Code Dim performance As New List(Of PerformanceCounter)
Dim CPUload As New List(Of Label)
Dim ProgressBar As New List(Of ProgressBar)
Dim locationx As Integer = 12
Dim locationy As Integer = 9

Dim WithEvents Timer As New Timer With {.Interval = 500, .Enabled = True}

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    Me.Size = New Size(300, 95)

    For i As Integer = 0 To System.Environment.ProcessorCount - 1

        Dim processor As New PerformanceCounter
        processor.CategoryName = "Processor"
        processor.CounterName = "% Processor Time"
        processor.InstanceName = i

        performance.Add(processor)

        Dim CPU As New Label
        CPU.Text = "Logical Processor #" + (i + 1).ToString
        CPU.AutoSize = True
        CPU.Location = New Point(locationx, locationy)

        Dim Progress As New ProgressBar
        Progress.Location = New Point(locationx, locationy + 16)
        Progress.Width = 221

        ProgressBar.Add(Progress)

        Dim Load As New Label
        Load.Text = "0%"
        Load.AutoSize = False
        Load.Size = New Size(33, 13)
        Load.Location = New Point(locationx + 227, locationy + 21)
```

```
CPULoad.Add(Load)
```

```
locationy += 51  
Me.Height += 45
```

```
Me.Controls.AddRange({CPU, Progress, Load})
```

```
Next  
End Sub
```

```
Private Sub TimerTick(ByVal sender As Object, ByVal e As EventArgs) Handles Timer.Tick  
    For i As Integer = 0 To performance.Count - 1  
        If ProgressBar(i).Value < performance(i).NextValue Then  
            ProgressBar(i).Value += 1  
        Else  
            Try  
                ProgressBar(i).Value -= 1  
            Catch ex As Exception : End Try  
        End If  
        CPULoad(i).Text = ProgressBar(i).Value.ToString + "%"   
    Next  
End Sub
```

That'll work if you through it into a Windows Form Application....Most likely.

It could be improved via fixing the Form Sizing (It's incredibly rough, i didn't do the math correctly, so unless the computer has 8 logical processors like mine, it won't look exactly right...), and instead of using 3 separate Lists of controls, possibly creating a single list of a custom class, which holds all the information related to the performance counter.

There also seems to be a problem with large CPU Activity Spikes (For example, i was going to provide a screenshot while i was encoding a video to make my CPU Activity spike, however it was essentially all 8 logical processors at 99% or 100%...), taking too long to update
Why would anyone need more than one CPU?

Nikey646 - January 28, 2014, 1:07 pm

(Jan 28, 2014 12:03 PM)ecaep42 Wrote: [\[CITE\]](#) Why would anyone need more than one CPU?
I have a single quad core CPU, the problem is, It has 8 Logical Processors....The 2 examples shown here only keep track of a static amount, meaning that i wouldn't be able to see all the activity in my CPU...

webdev77 - January 28, 2014, 1:11 pm

(Jan 28, 2014 12:03 PM)ecaep42 Wrote: [\[CITE\]](#) Why would anyone need more than one CPU?
Ever heard of a workstation? Dual Intel Xeons are a standard to go along with professional graphics. A single CPU doesn't cut it for complex 3D modeling work. An assembly of a few hundred parts will keep both processors at around 80% use pretty consistently.

ecaep42 - January 29, 2014, 3:20 am

(Jan 28, 2014 02:11 PM)webdev77 Wrote: [\[CITE\]](#) Ever heard of a workstation? Dual Intel Xeons are a standard to go along with professional graphics. A single CPU doesn't cut it for complex 3D modeling work. An assembly of a few hundred parts will keep both processors at around 80% use pretty consistently.
I haven't. It's practically used for a game develop workplace?

