

Forum: [Hack Forums](#)

Thread: [For Loop / MultiThreading Off ComboBox Value?](#)

Gorilla - February 18, 2017, 12:32 pm

I'm trying to implement a simple code but I want it to be threaded as many times as a combobox says.

For example:

Combo box value is: 25

This means the thread count is '25'.

I want a certain piece of code to run 25 times off a listbox but not duplicate each item 25 times like it seems to do when I 'multi-thread' it.

My idea for a code:

```
Code for i as integer = 1 to combobox1.value
listbox2.items.add(i)
next
end sub
```

end class

But how do I get this above code to run the combo box value into threads?

My code I'm trying to get multithreaded by the combobox value:

```
Code Private Sub Checker()
    Dim myproxy As WebProxy
    Dim proxies As New List(Of String)
    For Each proxy As String In ListBox1.Items
        Try
            myproxy = New WebProxy(proxy)
            Dim r As HttpWebRequest = WebRequest.Create("http://www.google.com")
            r.UserAgent = "Mozilla/5.0 (windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/29.0.1547.2 Safari/537.36"
            r.Timeout = 3000
            r.Proxy = myproxy
            Dim re As HttpWebResponse = r.GetResponse()
            ListBox2.Items.Add(proxy)
        Catch ex As Exception
            ListBox3.Items.Add("FAILED " & proxy)
        End Try
    Next
End Sub
```

(this is just a sample code and will not be used in my project).

I can't seem to figure out in my head how to make the two work off each other... Ninja tried helping

me, but I don't want to mooch off his quality skill.

Thanks for the help!

Ninja - February 18, 2017, 2:03 pm

So what is the problem with this code? Does it give an error?

jujuser - February 18, 2017, 2:07 pm

Happy to help, but need more info.

Do you want to run Checker() on a separate thread from the GUI-Main thread?
Is that what you are saying?

Use async maybe?

Reply with more details

Gorilla - February 18, 2017, 2:19 pm

I have fixed up the code to where it will run on it's own fairly quickly (based on thread count according to the combo box's value.)

But I'm having the issue to where it will check the same listbox item over and over until the threads are done, then move onto the next.

So if I set my thread count to 25 it will check the same listbox item 25 times before moving to the next item. Is there maybe a way to check if it's being used already?

My code :

Code 'Start Checker Thread

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    Dim threadcount As Integer = ComboBox1.Text
    Dim i As Integer = 0
    Do While (i <= threadcount)
        Dim thread As New Thread(New ThreadStart(AddressOf Checker))
        thread.IsBackground = True
        thread.Start()
        i += 1
    Loop
End Sub
```

Ninja - February 19, 2017, 1:22 am

What you should do is either write a class that distributes the workload over the threads, but that might be too complicated for you.

Or you can put a for loop inside the checker sub and have an integer for the index as a parameter and start the loop at that number, and increment with that number.

So something like this:

```

Code Do While (i <= threadcount)
    Dim thread As New Thread(Sub() Checker(i))
    thread.IsBackground = True
    thread.Start()
    i += 1
Loop

```

```

Private Sub Checker(int i)
    For meme As Integer = i To Proxies.Count Step i
        string proxy = Proxies(i)
    Next
End Sub

```

xxdarklord65xx - February 19, 2017, 1:48 am

(Feb 18, 2017 03:19 PM)Gørilla Wrote: [CITE](#) I have fixed up the code to where it will run on it's own fairly quickly (based on thread count according to the combo box's value.)

But I'm having the issue to where it will check the same listbox item over and over until the threads are done, then move onto the next.

So if I set my thread count to 25 it will check the same listbox item 25 times before moving to the next item. Is there maybe a way to check if it's being used already?

My code :

```

Code 'Start Checker Thread
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    Dim threadcount As Integer = ComboBox1.Text
    Dim i As Integer = 0
    Do While (i <= threadcount)
        Dim thread As New Thread(New ThreadStart(AddressOf Checker))
        thread.IsBackground = True
        thread.Start()
        i += 1
    Loop

```

End Subi have tried to do it . its really complicated . i know what did u mean . specially in brute force . like i was trying to make it 90 threads to finish the list so fast . but not like 90 threads on the same thing ..So what i did is that i made my http web request more faster using some techniques for example :

```

Code 'InCrease And Control Speed
    ServicePointManager.UseNagleAlgorithm = False
    ServicePointManager.Expect100Continue = False
    ServicePointManager.CheckCertificateRevocationList = False
    ServicePointManager.DefaultConnectionLimit =

```

ServicePointManager.DefaultNonPersistentConnectionLimitit made my application goes faster . but ofcourse i was having a good internet it was my main reason making it faster . my result was like 120 name checked per 5 seconds . And yes i forget to mention . try to use USING END USING and use Stream to dispose at the last .. i tried to gave some informations for you mate , Good luck

Closed User - February 19, 2017, 4:30 am

I would advice you to look into Parallel programming :)

Cairns - February 19, 2017, 5:10 am

Your problem is you're running the same method over and over. You are going to have to be smarter about this when it comes to multi threading

xxdarklord65xx - February 19, 2017, 6:27 am

(Feb 19, 2017 05:30 AM)Big Benim Wrote: [\[CITE\]](#) I would advice you to look into Parallel programming :)
true . its all about the parallel threading .

Ninja - February 19, 2017, 6:42 am

If you want you can use the Partitioner class to partition a collection into an amount of IEnumerators and then use a while loop to loop over the proxies.

Gørilla - February 19, 2017, 11:53 am

Ninja helped me find my problem which is my start checker sub is looped. But if I remove my loop, I have to remove my do while. So if I were to remove the do while, would I just write it out plain without any 'do' statements? I'm also a beginner in multithreading and this is just to help me learn.

If I try to just write it out with `i <= threadcount` it throws a syntax error on `i` .

I feel like I'm so close! Now I'm having an issue where a variable is being hidden in a block or something..

Screenshot: <https://gyazo.com/85c539d4dd1489d083415df82a518ee8>

Anyone else able to help me? I really need it!

Decim - February 20, 2017, 7:57 am

(Feb 19, 2017 12:53 PM)Gørilla Wrote: [\[CITE\]](#) Ninja helped me find my problem which is my start checker sub is looped. But if I remove my loop, I have to remove my do while. So if I were to remove the do while, would I just write it out plain without any 'do' statements? I'm also a beginner in multithreading and this is just to help me learn.

If I try to just write it out with `i <= threadcount` it throws a syntax error on `i` .

I feel like I'm so close! Now I'm having an issue where a variable is being hidden in a block or something..

Screenshot: <https://gyazo.com/85c539d4dd1489d083415df82a518ee8>

Anyone else able to help me? I really need it!

You're declaring the variable 'i' twice.

? ?

Closed User - February 20, 2017, 8:39 am

(Feb 19, 2017 12:53 PM)Gørilla Wrote: [\[CITE\]](#) Ninja helped me find my problem which is my start checker sub is looped. But if I remove my loop, I have to remove my do while. So if I were to remove the do while, would I just write it out plain without any 'do' statements? I'm also a beginner in multithreading and this is just to help me learn.

If I try to just write it out with `i <= threadcount` it throws a syntax error on `i` .

I feel like I'm so close! Now I'm having an issue where a variable is being hidden in a block or something..

Screenshot: <https://gyazo.com/85c539d4dd1489d083415df82a518ee8>

Anyone else able to help me? I really need it!

PARALLEL PROGRAMMING YA MONKEY!

On the real, look into parallel threading and you won't ever have those issues

Gørilla - February 20, 2017, 12:30 pm

Ok, I'll look into it. Thanks for the help guys!