

Forum:[Hack Forums](#)

Thread: [Cycle through Text Files For Specific Line?](#)

Haters Ganna Hate - September 28, 2017, 8:39 pm

So I have a very large 10GB text file that I split into 130 80MB text files. I need to cycle through all of them until I find a certain name.

I know by looking at one text file, I can use this:

Code Using reader As New StreamReader("Test.txt")

While Not reader.EndOfStream

Dim line As String = reader.ReadLine()

If line.Contains("Banana") Then

TextBox1.Text = line

Exit While

End If

End While

End Using

But I need it to go to the next text file if the first text file does not have it. Luckily, the text files all start with the name "split" and their names are like:split1

split2

split3

I just need a loop or something that checks every single "split" text file for a keyword until it found it.

Got a code example?

DBots - September 28, 2017, 10:14 pm

[Directory.GetFiles\(\)](#) has an option to set a filter.

Here's what I did:

Haters Ganna Hate - September 29, 2017, 9:10 am

(Sep 28, 2017 10:14 PM)DBots Wrote: [\[CITE\] Directory.GetFiles\(\)](#) has an option to set a filter.

Here's what I did:

Does not seem to work. Here is my Form:

And here is my code: <https://pastebin.com/w94pEey4>

I have two files called "split1" which contains:

Quote:John Smith

John Rocke

Jacob Means

Rachial Sam

And "split2" which contains:

Quote:Naing Man

Wu Sam Lem
Kemo Kam
Haha Rob

But when I search up the term "Haha Rob", it says it finds it in both split1 and split2 repeatedly. Im a little tired now and its dark so maybe I am not seeing something?

DBots - September 29, 2017, 9:52 am

after you .ReadLine, you need to compare it to the target string like
If sLine.Equals(sTargetString) Then..

So yeah it seems you overlooked that part which you actually had something like that in your original code.

Owl - September 29, 2017, 10:41 am

Code Dim target As String = "Haha Rob"

```
For Each file In Directory.GetFiles("path", "split*.txt")
    Using sr As New StreamReader(file)
        While Not sr.EndOfStream
            Dim line As String = sr.ReadLine()

            If line.Contains(target) Then
                Console.WriteLine(line)

            Exit For
        End If
    End While
End Using
Next
Works fine.
```

DBots - September 29, 2017, 10:59 am

OrKa got it! But man you should have let him try a little more he almost had it lol.

joey karate - September 29, 2017, 11:00 am

open up notepad++
<http://prntscr.com/grabni>

dicktembe - September 30, 2017, 8:38 am

I cycle through text files like this:

```
Code If IO.File.Exists(datafilepath) Then
    Dim datafile As List(Of String) = IO.File.ReadAllLines(datafilepath).ToList()
    For i = 0 To datafile.Count - 1
        If datafile(i).Contains("SEARCH STRING HERE") Then
            'do something here
        End If
    Next
End If
```

Example, here I open a file and get to a certain line that contains 'MouseSensitiveList=', I then get values of certain settings and replace them because I know they are 8 chars long. However you can do anything you like:

```
Code If IO.File.Exists(configpath) Then
    Dim config As List(Of String) = IO.File.ReadAllLines(configpath).ToList()
```

```

For i = 0 To config.Count - 1
    If config(i).Contains("MouseSensitiveList=") Then
        For j = 0 To 8
            Dim indexpos As Integer
            If j = 0 Then
                indexpos = config(i).IndexOf("LastConvertedMouseSensitivity=") + 30
            Else
                indexpos = config(i).IndexOf("LastConvertedMouseSensitivity=", indexpos) +
30
            End If
            If j < 3 Then
                config(i) = config(i).Remove(indexpos, 8).Insert(indexpos, txtPUBG.Text)
            ElseIf j >= 3 And j < 8 Then
                config(i) = config(i).Remove(indexpos, 8).Insert(indexpos,
txtPUBGSCOPED.Text)
            End If
        Next
        IO.File.WriteAllLines(configpath, config)
        MsgBox("Config file written! (I hope, hehe xd)")
    End If
Next
End If

```

Ninja - October 1, 2017, 11:01 am

(Sep 30, 2017 08:38 AM)dicktembe Wrote: [\[CITE\]](#) - SNIP -

You read all contents of the text file at once into the program. This is not a good solution at all for handling large text files like OP is (130x 80MB).

@OP; if you need random access I suggest using the [MemoryMappedFile](#) class, so you can get complete, and random access to a file without the need for seeking. If you don't, then a StreamReader and reading line by line like you already have is fine.

Cairns - November 7, 2017, 8:43 am

If you're reading a 10GB file that wont go too smoothly with any of those methods.

I suggest you read x amount of lines (5000 lines for example) and append it to a stringbuilder, then check the stringBuilder.ToString if it contains the line text. If not, then continue running the cycle/loop until done. Do not read a 10GB file into memory at once and continuously check EACH line because it's gonna be a memory killer and its gonna take forever.

gigajew - November 7, 2017, 9:30 am

Here's how I do it

```

Code using (var reader = new StreamReader("file.txt")) {
    while (sr.Peek() >= 0) {
        var line = sr.ReadLine();
        Console.WriteLine(line); // do whatever you want with the line
    }
}

```

0xDEAD10CC - November 7, 2017, 2:10 pm

None of the above solutions is going to work very fast without dealing with mapping files into memory and reading things that way. When you start dealing with GB's the naive and most obvious approaches are the worst. That class posted by Ninja is essentially a wrapper around native

functions like `MapViewOfFile()`, which is what I personally use to read large files > 1GB in fractions of the time it takes using normal (portable) methods. (`mmap()` for those on Linux.)

That class makes it much easier than using the functions directly though because otherwise you'll be calculating pointer offsets, allocating to match system granularity, among other things.