

Re: How does "new" work in a loop?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-07/msg00808.html>

- *From:* "Matt" <matttelles@xxxxxxxxxxx>
 - *Date:* 6 Jul 2006 14:00:49 -0700
-

Tony Sinclair wrote:

I'm just learning C#. I'm writing a program (using Visual C# 2005 on WinXP) to combine several files into one (HKSplit is a popular freeware program that does this, but it requires all input and output to be within one directory, and I want to be able to combine files from different directories into another directory of my choice).

My program seems to work fine, but I'm wondering about this loop:

```
for (int i = 0; i < numFiles; i++)
{
    // read next input file

    FileStream fs = new FileStream(fileNames[i],
    FileMode.Open, FileAccess.Read, FileShare.Read);
    Byte[] inputBuffer = new Byte[fs.Length];

    fs.Read(inputBuffer, 0, (int)fs.Length);
    fs.Close();

    //append to output stream previously opened as fsOut

    fsOut.Write(inputBuffer, 0, (int) inputBuffer.Length);
    progBar.Value++;
} // for int i
```

As you can see, the objects fs and inputBuffer are both created as "new" each time through the loop, which could be many times. I didn't think this would work; I just tried it to see what kind of error message I would get, and I was surprised when it ran. Every test run has produced perfect results.

So what is happening here? Is the memory being reused, or am I piling up objects on the heap that will only go away when my program ends, or am I creating a huge memory leak?

Re: How does "new" work in a loop?

Unlikely that you are creating a memory leak. C# uses garbage collection.

When the object goes out of scope (in your case, the } marked // for int i) the object is destroyed. The next time through the loop, a new one is created.

Matt

.