

Re: How is this conditional able to execute?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2005-04/msg02991.html>

- *From:* "Stephany Young" <noone@localhost>
 - *Date:* Tue, 19 Apr 2005 00:11:06 +1200
-

This is getting scary :)

Lets take this one step at a time.

I'm a bit perplexed as to why a receive buffer would be such a strange size. Could it be that the value of `tcpClient.ReceiveBufferSize` is actually 8192 (8KB)? If so, then the array declaration should be:

```
Dim bytes(tcpClient.ReceiveBufferSize - 1) As Byte
```

In VB.NET, arrays are 0 based and the value specified is the upper bound of the array as opposed to the number of elements (as in C#.NET).

Can you dump the first 50 bytes of the array on the first call. Directly after the line:

```
networkStream.Read(bytes, 0, tcpClient.ReceiveBufferSize)
```

insert:

```
For _i As Integer = 0 to 48  
Console.Write("{0},", bytes(_i).ToString())  
Next  
Console.WriteLine("{0}", bytes(_i).ToString())
```

Add another line:

```
Console.WriteLine()
```

and place a breakpoint on this line.

Run the code then copy and paste the result into a post to this thread.

It will be interesting to see just what characters are arriving.

A new byte array is created each time the `networkStream` object is read so there is nothing from the previous read remaining. Therefore the comment about the 'values shifting' is a red herring. What you are getting is

Re: How is this conditional able to execute?

actually what is coming from the networkStream object.

Also, is there any particular reason why you are using UTF8 encoding to interpret the data? Is it specified somewhere that the data arriving in the networkStream object is actually UTF8 or is just a stab in the dark?

```
"Brett" <no@xxxxxxx> wrote in message
news:%23aSOEDARFHA.2744@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>I tried this but it isn't working either. At one point, the bytes array
>has non zero elements up to element 35. returndata has the value of "" (or
>so it seems) and length of 8193, same as the bytes array. Through each
>pass of the loop, the non zero elements increase in the bytes array. So
>it goes from element 35 to 36, 37 and so on. The values are shifting so
>that the first 10 are now zero. Then the first 11, 12, 13 and so on.
>
> Any other suggestions?
>
> Thanks,
> Brett
>
> "Stephany Young" <noone@localhost> wrote in message
> news:u1TIDb9QFHA.2680@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>> Sorry, my mistake. Null (Chr(0)) is not treated as whitespace by the
>> String.Trim Method.
>>
>> Instead, you can use one of the overloaded String.Trim methods.
>>
>> Dim returndata As String =
>> Encoding.UTF8.GetString(bytes).Trim(Char.MinValue)
>>
>> Char.MinValue is a constant for 0x00 which equates to Null or Chr(0).
>>
>> This overload removes all occurrences of a set of characters specified in
>> an array from the beginning and end of the string instance.
>>
>>
>> "Brett" <no@xxxxxxx> wrote in message
>> news:uPLcAZ6QFHA.3988@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>>>
>>> "Stephany Young" <noone@localhost> wrote in message
>>> news:%23hNf834QFHA.3988@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>>>> What test are you using to determine that returndata.ToString = ""?
>>>>
>>>> Are you , for example, executing a:
>>>>
>>>> Console.WriteLine("returndata.ToString = " & returndata.ToString)
>>>>
>>>> and getting the result:
>>>>
>>>> returndata.ToString =
```


Re: How is this conditional able to execute?

- ◆ **Re: How is this conditional able to execute?**
 - ◇ From: Brett
- ◆ **Re: How is this conditional able to execute?**
 - ◇ From: Stephany Young
- ◆ **Re: How is this conditional able to execute?**
 - ◇ From: Brett
- ◆ **Re: How is this conditional able to execute?**
 - ◇ From: Stephany Young
- ◆ **Re: How is this conditional able to execute?**
 - ◇ From: Brett

- Prev by Date: **RE: Windows messages**
- Next by Date: **Re: Windows messages**
- Previous by thread: **Re: How is this conditional able to execute?**
- Next by thread: **Re: How is this conditional able to execute?**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**