

The results are in! See what nearly 90,000 developers picked as their most loved, dreaded, and desired coding languages and more in the 2019 Developer Survey.

What is the easiest way in C# to trim a newline off of a string?

[Ask Question](#)

I want to make sure that `_content` does not end with a `NewLine` character:

97



```
_content = sb.ToString().Trim(new char[] { Environment.NewLine });
```



but the **above code doesn't work** since `Trim` seems to not have an overloaded parameter for a collection of strings, only characters.

9

What is the simplest one-liner to remove an `Environment.NewLine` from the end of a string?

c#

string

asked Jun 24 '09 at 12:22



[Edward Tanguay](#)

80.7k 271 646 982

3 btw, you probably want `TrimEnd` – [Marc Gravell](#) ♦ Jun 24 '09 at 12:28

10 Answers

216

The following works for me.

```
sb.ToString().TrimEnd( '\r', '\n' );
```

or

```
sb.ToString().TrimEnd( Environment.NewLine.ToCharArray());
```

answered Jun 24 '09 at 12:30



Simon Wilson

6,838 3 23 23

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams

Q&A for work



Learn More

very concise and deals with the \r\n issue, nice – [Edward Tanguay](#) Jun 24 '09 at 13:28

- 1 While trimming ANY combination of characters that make up a new line char(s) does address underlying problem and the spirit of the question, it also leaves a potential for unwanted functionality in the general sense. In a file processing senario I could see a need for leaving the first \n intact in a string of HeresALineForUnixToBeFedFromWindows\r\n\r\n. While more verbose, John Skeet's answer is probably the best and Scott Weinstein's ReadLine is also more "accurate" (albeit maybe with a bit more overhead) – [b_levitt](#) Jan 14 '13 at 23:07

- 22 TrimEnd() is already doing the Job without any Parameters! msdn.microsoft.com/de-de/library/... and msdn.microsoft.com/de-de/library/... – [Markus](#) Mar 31 '14 at 9:42

- 2 @Markus Thanks I found this because I do not want this behavior but was experiencing it. I was under the impression \n was not a white space but it is and .TrimEnd and .Trim clear all [white space](#) characters natively – [David Carrigan](#) May 13 '15 at 20:46

.Trim() removes \r\n for me (using .NET 4.0).

edited Sep 14 '17 at 16:25

29

answered May 6 '13 at 18:44



xr280xr

6,423 3 54 92

This does the job. Least code too. – [JumpingJezza](#) Dec 16 '14 at 4:31

0

As Markus pointed out TrimEnd is doing the job now. I needed to get line feeds and white space from both ends of string in Windows Phone 7.8 environment. After having chased different more complex options my problem was solved by using Trim() only - passed the following tests nicely

```
[TestMethod]
[Description("TrimNewLines tests")]
public void Test_TrimNewLines()
{
    Test_TrimNewLines_runTest("\n\r      testi   \n\r", "testi");
    Test_TrimNewLines_runTest("\r      testi   \r", "testi");
    Test_TrimNewLines_runTest("\n      testi   \n", "testi");
    Test_TrimNewLines_runTest("\r\r\r\r\r\n\r      testi   \r\r\r\r\r\n\r", "testi");
    Test_TrimNewLines_runTest("\n\r \n\n\n\n     testi ääl.,\n\rääl.");
    Test_TrimNewLines_runTest("\n\n\n\n        testi ja testi \n\nja testi");
    Test_TrimNewLines_runTest("", "");
    Test_TrimNewLines_runTest("\n\r\n\n\r\n\r\n", "");
    Test_TrimNewLines_runTest("\n\r \n\n \n\n", "");
}

private static void Test_TrimNewLines_runTest(string _before, string _expected)
{
    string _response = _before.Trim();
    Assert.IsTrue(_expected == _response, "string '" + _before + "'  
to '" + _response + "' - should have been '" + _expected + "'");
}
```

answered Oct 16 '14 at 5:00



juhariis

335 3 11

How about:

14

```
public static string TrimNewLines(string text)
{
    while (text.EndsWith(Environment.NewLine))
    {
        text = text.Substring(0, text.Length - Environment.NewLine.Length);
    }
    return text;
}
```

It's somewhat inefficient if there are multiple newlines, but it'll work.

Alternatively, if you don't mind it trimming (say) `"\r\r\r\r\r"` or `"\n\n\n\n"` rather than just `"\r\n\r\n\r\n\r\n"` :

```
// No need to create a new array each time
private static readonly char[] NewLineChars = Environment.NewLine.ToCharArray();

public static string TrimNewLines(string text)
{
    return text.TrimEnd(NewLineChars);
}
```

edited Jul 10 '13 at 3:55

user1968030

answered Jun 24 '09 at 12:27



Jon Skeet

1098k 697 7996

8478

Probably the most accurate solution, but not a one-liner :-) – [bang](#) Jun 24 '09 at 12:35

I had to remove the new lines all over the text. So I used:

-2

```
while (text.Contains(Environment.NewLine))
{
    text = text.Substring(0, text.Length - Environment.NewLine.Length);
}
```

answered Sep 15 '11 at 11:41



[Ekaterina](#)

189 1 4 22

1 This will be a disaster if text contains NewLine(s) that is not in the end. – [Stefan Bergfeldt](#) Nov 22 '17 at 14:40

1

Somewhat of a non-answer, but the easiest way to trim a newline off of a string is to not have the newline on the string in the first place, by making sure it is never seen by your own code. That is, by using native functions which remove the newline. Many stream and file/io methods will not include the newline if you ask for output line by line, though it may be necessary to wrap something in a `System.IO.BufferedStream`.

Things like `System.IO.File.ReadAllLines` can be used in place of `System.IO.File.ReadAllText` most of the time, and `ReadLine` can be used instead of `Read` once you are working with the right type of stream (e.g. `BufferedStream`).

answered Jun 24 '09 at 13:21



Brian

19k 13 66 152



How about just:

2

```
string text = sb.ToString().TrimEnd(null)
```



That will pull all whitespace characters from the end of the string -- only a problem if you wanted to preserve non-newline whitespace.

answered Jun 24 '09 at 12:34



Richard Dunlap

1,752 9 14



```
_content = sb.TrimEnd(Environment.NewLine.ToCharArray());
```

4



This will of course remove "\r\r\r\r" as well as "\n\n\n\n" and other combinations. And in "enviroments" where NewLine is other than "\n\r" you might get some strange behaviors :-)

But if you can live with this then I belive this is the most effectiv way to remove new line characters at the end of a string.

edited Jun 24 '09 at 12:33



bang

3,482 16 23

Use the Framework. The `ReadLine()` method has the following to say:

8

A line is defined as a sequence of characters followed by a line feed ("`\n`"), a carriage return ("`\r`") or a carriage return immediately followed by a line feed ("`\r\n`"). The string that is returned does not contain the terminating carriage return or line feed.

So the following will do the trick

```
_content = new StringReader(sb.ToString()).ReadLine();
```

answered Jun 24 '09 at 12:31



[Scott Weinstein](#)

14.6k 9 66 110

- 1 Elegant yes, but not easy to understand for others reading this line. Needs an explaining comment (in the same line of course to remain a one-liner :)) – [gumo](#) Dec 20 '13 at 15:47

What about

5

```
_content = sb.ToString().Trim(Environment.NewLine.ToCharArray());
```

answered Jun 24 '09 at 12:25



[heavyd](#)

14.4k 2 49 67

That doesn't do *quite* the same thing - it'll trim "`abc\n\n\n\n\n`" for example,

even if Environment.NewLine is crlf – [Marc Gravell](#) ♦ Jun 24 '09 at 12:28

Declare it as a feature :-) I like the solution. – [Stefan Steinegger](#) Jun 24 '09 at 12:37
