Psychic debugging: Why can’t StreamReader read apostrophes from a text file?

devblogs.microsoft.com/oldnewthing/20080811-01

August 11, 2008

Raymond Chen

As is customary, the first day of CLR Week is a warm-up. Actually, today’s question is a BCL question, not a CLR question, but only the nitpickers will bother to notice.

> Can somebody explain why StreamReader can’t read apostrophes? I have a text file, and I read from it the way you would expect:
>
> ```
> StreamReader sr = new StreamReader(“myfile.txt”);
> Console.WriteLine(sr.ReadToEnd());
> sr.Close();
> ```
>
> I expect this to print the contents of the file to the console, and it does—almost. Everything looks great except that all the apostrophes are gone!

You don’t have to have very strong psychic powers to figure this one out.

Here’s a hint: In some versions of this question, the problem is with accented letters.

Your first psychic conclusion is that the text file is probably an ANSI text file. But StreamReader defaults to UTF-8, not ANSI. One version of this question actually came right out and asked, “Why can’t StreamReader read apostrophes from my ANSI text file?” The alternate version of the question already contains a false hidden assumption: StreamReader can’t read apostrophes from an ANSI text file because StreamReader (by default) doesn’t read ANSI text files at all!

But that shouldn’t be a factor, since the apostrophe is encoded the same in ANSI and UTF-8, right?

That’s your second clue. Only the apostrophe is affected. What’s so special about the apostrophe? (The bonus hint should tip you off: What’s so special about accented letters? What property do they share with the apostrophe?)

There are apostrophes and there are apostrophes, and it's those "weird" apostrophes that are the issue here. Code points U+2018 (‘) and U+2019 (’) occupy positions 0x91 and 0x92, respectively, in code page 1252, and these "weird" apostrophes are all illegal lead bytes in UTF-8 encoding. And the default behavior for the Encoding.UTF8Encoding encoding is to ignore invalid byte sequences. Note that StreamReader does not raise an exception when incorrectly-encoded text is encountered. It just ignores the bad byte and continues as best it can, following Burak's advice.

Result: StreamReader appears to ignore apostrophes and accented letters.

There are therefore multiple issues here. First, you may want to look at why your ANSI text file is using those weird apostrophes. Maybe it's intentional, but I suspect it isn't. Second, if you're going to be reading ANSI text, you can't use a default StreamReader, since a default StreamReader doesn't read ANSI text. You need to set the encoding to `System.Text.Encoding.Default` if you want to read ANSI text. And third, why are you using ANSI text in the first place? ANSI text files are not universally transportable, since the ANSI code page changes from system to system. Shouldn't you be using UTF-8 text files in the first place?

At any rate, the solution is to decide on an encoding and to specify that encoding when creating the StreamReader.

This exercise is just another variation on Keep your eye on the code page.

Raymond Chen

Follow