

Re: \n?? \r??

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-09/4327.html>

From: James Curran (JamesCurran_at_mvps.org)

Date: 09/17/04

Date: Fri, 17 Sep 2004 12:20:13 -0400

"Sarah" <Sarah@discussions.microsoft.com> wrote in message
news:C4BEF541-4382-4659-AFE0-F0DC49A9A7C1@microsoft.com...

> *what's the difference between \n and \r*

To understand the characters, you need to know a little history. When the original ASCII standard was defined, computers used teletypes for interacting with users --- Video monitors were rare to nonexistent. Moving to a new line was a two step process: you had to move the carriage (the thing stamping letters on the paper) back to the left margin, and you had to advance the paper up one line. When the control codes were defined, it was decided that these actions would be given separate codes, for more flexible control over printing. For one thing, this allowed over- printing -- returning the carriage without advance to paper, so the next line is printed on top of the last. The most common use for this was to obscure a password the user had to enter, so it couldn't be read by someone pulling an old listing out of the garbage. So, char 10 (0x0A) is defined as a "Line Feed" (LF), and char 13 (0x0D) is defined as "carriage return" (CR)

As CRT terminals started to be used, separate codes was less important, but hard disk space was at a premium, so OS designer started to use the convention of using just one of the characters as an "End of Line" code. Most people used CR; UNIX, in their iconoclastic way, chose LF instead. When the C language was defined --- by Unix users --- they made LF the "next line" character ("\n"), and throw in CR ("\r") as a sop to everyone else. (Those symbols were carried over to C++ & C#)

Then in 1982, IBM announced it's PC, and, at the same time, finally decided to embrace ASCII (they had been using the very bizarre EBCDIC before), --- and really embraced it and actually required both CR LF characters as an end-of-line, as it said in the official ignored standard. Most of the systems using just CR eventually converted to CR LF, or just faded away. Unix, on the other hand, continued using just LF.

So, now (c1985) we have a programming language (C), producing files with just a LF, and an operating system (MS-DOS) requiring CR LF. So, the

microsoft.public.dotnet.languages.csharp: Re: \n?? \r??

authors of C Run-time library created the concept of "text mode" files, where, when reading files it would convert CR LF into LF, and when writing files it would convert LF to CR LF. (Similarly, this has been carried over of C++'s iostreams & to the .Net CLR)

--

Truth,

James Curran

Home: www.noveltheory.com Work: www.njtheater.com

Blog: www.honestillusion.com Day Job: www.partsearch.com

(note new day job!)