

---

Subject: Re: Socket - send multiple lines  
Posted by [nneilson](#) on Sat, 20 Nov 2010 22:46:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Honza: After a little research I understand where "if(\*q=='\n')" can and cannot be used.

Reading a file for each char does read '\n' or whatever for different OS.

C++ "getline", Java "readline" or whatever for different languages is to get the line.

Some languages like C++ drops the '\n' (you don't see it when debugging) and replaces that with the required '\0' to terminate the string. If several lines are to be included in one string the '\0' needs to be replaced with '\n'. The '\0' needs to be retained or added at the end.

In Java the '\n' is read (you can see and manipulate it) but not included in the string unless the code specifically inserts it.

A string is a char buf. The difference on how it is referenced/used is just semantics.

What is sent through my socket is just a bunch of characters. My server code includes the parsing of the data received, if the server just received the data and another function did the parsing no changes to the server code would have been necessary.

I did get my socket server to work with the String/char buf that has a single line or multiple lines. The client just sends what it is told to send, no changes were required there.

---