



Download

A: It's because you're creating a new line on your file but not close it: if (ConnectionSocket.Connected) { // Grab the linefeed from the text file and add it to the end of the response text status = ConnectionSocket.EndReceive(bytesReceived); status = ConnectionSocket.BeginReceive(bytesReceived, 0, bytesReceived, 0, new AsyncCallback(ReadCallback), ConnectionSocket); } else status = SocketError.ReceiveFailure; if (!ConnectionSocket.Connected) { text += Environment.NewLine; if (textLength > 0) { text = text.Remove(0, textLength); } } if (!ConnectionSocket.Connected) status = SocketError.ReceiveFailure; You don't seem to read the file, just append the linefeed. You should close the file. The easiest way to do so in c# is: File.WriteAllText(path, text); Q: Run time-dependent dynamic sql update statement I've been playing around with sql insert/update statements, and have managed to come up with some routines I think will simplify the process of assigning the data being entered into a form to my databased (EF code first). I've decided to go for a single table for the initial form input then use dynamic sql to add the additional fields as data is entered by the user (details here). The bit I'm struggling with is how to dynamically generate the update statement that I will run. The update statement is generated at run-time based on the table and fields that are currently inserted. My original plan was to create a stored procedure to dynamically generate the update statement and pass that on to my EF code first context (the context is wrapped up in a transaction). I've looked into string concatenation, I can create the sql statement fine, but I'm a little stuck on how to build up a set of where clauses/aggregations/group clauses to run the update statement against (I don't want to have to manually enter them). I can write a function that concaten

