





Create a text.txt file and place it in a folder called FT

using System;

using System.Collections.Generic;

using System.Text;

using System.Net;

using System.Net.Sockets;

using System.IO;

namespace SendFileServer

{

class Program

{

static void Main(string[] args)

{

try

{

Console.WriteLine("That program can transfer small file. I've test up to 850kb file");

IPEndPoint ipEnd = new IPEndPoint(IPAddress.Any, 5656);

Socket sock = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.IP);

sock.Bind(ipEnd);

sock.Listen(100);

//clientSock is the socket object of client, so we can use it now to transfer data to client

Socket clientSock = sock.Accept();

string fileName = "test.txt"; // "Your File Name";

string filePath = @"I:\FT\"; //Your File Path;

byte[] fileNameByte = Encoding.ASCII.GetBytes(fileName); //not content just the name test.txt

byte[] fileData = File.ReadAllBytes(filePath + fileName); // Read content of :\FT\test.txt

byte[] clientData = new byte[4 + fileNameByte.Length + fileData.Length];

byte[] fileNameLen = BitConverter.GetBytes(fileNameByte.Length);

fileNameLen.CopyTo(clientData, 0);

fileNameByte.CopyTo(clientData, 4);

fileData.CopyTo(clientData, 4 + fileNameByte.Length);

clientSock.Send(clientData);

Console.WriteLine("File:{0} has been sent.", fileName);

clientSock.Close();

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine("File Receiving fail." + ex.Message);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Net;

using System.Net.Sockets;

using System.IO;

namespace SendFileClient

{

class Program

{

static void Main(string[] args)

{

try

{

Console.WriteLine("That program can transfer small file. I've test up to 850kb file");

IPAddress[] ipAddress = Dns.GetHostAddresses("localhost");

IPEndPoint ipEnd = new IPEndPoint(ipAddress[1], 5656);

Socket clientSock = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.IP);

clientSock.Connect(ipEnd);

byte[] clientData = new byte[1024 \* 5000];

string receivedPath = "I:/";

int receivedBytesLen = clientSock.Receive(clientData);

int fileNameLen = BitConverter.ToInt32(clientData, 0);

string fileName = Encoding.ASCII.GetString(clientData, 4, fileNameLen);

Console.WriteLine("Client:{0} connected & File {1} started received.", clientSock.RemoteEndPoint, fileName);

BinaryWriter bWrite = new BinaryWriter(File.Open(receivedPath + fileName, FileMode.Append)); ;

bWrite.Write(clientData, 4 + fileNameLen, receivedBytesLen - 4 - fileNameLen);

Console.WriteLine("File: {0} received & saved at path: {1}", fileName, receivedPath);

bWrite.Close();

clientSock.Close();

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine("File Sending fail." + ex.Message);

}

}

}

}

ASCIIEncoding.GetBytes Method

A string can be converted into a byte array. Strings are stored with two bytes per character. ASCII only allows one byte per character. This can cause data loss. With Encoding.ASCII.GetBytes, and GetString, we perform this conversion



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace EncodingASCII

{

class Program

{

static void Main(string[] args)

{

Byte[] bytes;

String chars = "ASCII Encoding Example";

ASCIIEncoding ascii = new ASCIIEncoding();

int byteCount = ascii.GetByteCount(chars.ToCharArray(), 6, 8);

bytes = new Byte[byteCount];

int bytesEncodedCount = ascii.GetBytes(chars, 6, 8, bytes, 0);

Console.WriteLine(

"{0} bytes used to encode string.", bytesEncodedCount

);

Console.Write("Encoded bytes: ");

foreach (Byte b in bytes) {

Console.Write("[{0}]", b);

}

Console.WriteLine();

Console.Read();

}

}

}

OPPOSITE:



byte[] array2 =

{

68,

111,

116,

32,

78,

101,

116,

32,

80,

101,

114,

108,

115

};

string value = ASCIIEncoding.ASCII.GetString(array2);

Console.WriteLine(value);

//Another Example

const string input = "Dot Net Perls";

// Invoke GetBytes method.

// ... You can store this array as a field!

byte[] array = Encoding.ASCII.GetBytes(input);

// Loop through contents of the array.

foreach (byte element in array)

{

Console.WriteLine("{0} = {1}", element, (char)element);

}

