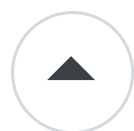


What's the best way to send a file over a network using C#?

Asked 16 years, 3 months ago Modified 12 years, 1 month ago

Viewed 35k times



8



Can anyone point me to a tutorial on the best way to open a connection from client to server, read in a binary file and send its contents reliably across the network connection? Even better, is there an open source library that already does this that I could refer to?



c#

.net

ftp

tcp

data-transfer



Share

Improve this question

Follow

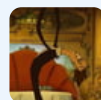
edited Nov 22, 2012 at 13:57



[Mike Pennington](#)

43k ● 21 ● 139 ● 188

asked Sep 18, 2008 at 18:19



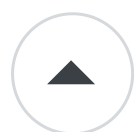
[screenglow](#)

1,886 ● 5 ● 21 ● 23

6 Answers

Sorted by:

Highest score (default)



You should look into binary serialization and sending it over a TCP socket.

6

Good explanation on different types of serialization:



<http://www.dotnetspider.com/resources/408-XML-serialization-Binary-serialization.aspx>



Good primer on TCP Client/Server in C#:

<http://www.codeproject.com/KB/IP/tcpclientserver.aspx>



Share Improve this answer

answered Sep 18, 2008 at 18:32

Follow



GEOCHET

21.3k ● 15 ● 77 ● 99

-
- 1 The second link is from 2001, which was old even when this question was fresh. It isn't much of a tutorial and is more of a code sample. – [YetAnotherRandomUser](#) Feb 18, 2018 at 22:00
-



2



This depends what you mean by network - if you're copying on a local network you can just use the file copy operations inside System.IO. If you're wanting to send to remote servers I do this using web services. I compress byte arrays and send them over and decompress on the remote side. The byte array is super easy to write back to disk using streams.



I know some people prefer base 64 strings instead of the byte[]. not sure if it matters.

Share Improve this answer

answered Sep 18, 2008 at 18:26

Follow



typemismatch

2,058 ● 3 ● 23 ● 26



2



Sockets may be the best route if you're just having to do it over the network. If you use TCP, you get the reliability of communication but take an impact on speed. If you need higher performance, you could try using UDP instead. But the downside to UDP is that packet delivery and order is not guaranteed, so you would need to write all that plumbing yourself.

If you are needing to transfer files over the web itself (programmatically, and if you can't use FTP), then a web service approach via [MTOM](#) might fit your needs.

If you are building on top of Windows Server 2003 R2, Windows Vista, or Windows Server 2008 and doing internal network transfers, another option is to leverage the new [Remote Differential Compression](#) feature. This not only does a really good job at compressing a file to minimize network traffic, but is also used directly by DFS replication. Downside (as a .NET developer), it's a COM+ technology.

Share Improve this answer

answered Sep 18, 2008 at 19:47

Follow



[Jason Olson](#)

3,696 ● 2 ● 24 ● 24



1

I wouldn't use HTTP or FTP, for a single file it's too much overhead and too much to code, especially having a simple TCP server almost already made for you in C#.



Share Improve this answer

Follow

answered Sep 18, 2008 at 18:26



Guille

63 ● 2 ● 5 ● 10



How about using HTTP or FTP? They were sort of made for this.

0

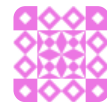
Alex



Share Improve this answer

Follow

answered Sep 18, 2008 at 18:23



Alex Fort

18.8k ● 5 ● 44 ● 51



Depending upon where you are sending the file to, you might want to take a look at [WebClient.UploadFileAsync](#) and [WebClient.UploadFile](#).

0



Share Improve this answer

Follow

answered Sep 18, 2008 at 18:23



rjzii

14.5k ● 12 ● 81 ● 122

