Network Programming made easy!

www.protocol-builder.com
In most network applications we have a server and some clients want to connect to it to exchange some kind of data ...
Clients connect to the server using its IP or name and the application port, the listener in the server will answer the clients at the first instance....
The Listener initializes the connections with the client connection objects and spawns new thread for each client, these threads will execute the code of the server connection objects...
Each Server Connection communicates with its client using new random port and exchanges data as required.
All Server Connections log their activities using the same logger object, each client connection will have its own logger, and the listener will have its own logger as well.
Now on, client and server connections will manage the connection and they will be responsible for dropping it, although the Listener has the ability to abort all server connections in some special cases depending on the application scenario.
From Programming point of view, we will have Client Connection Class, Server connection Class, Listener Class, Logger Class, and classes to represent all kinds of packets ...
Request packets are sent from client to server only; we can send one or more request packets to the server to get one or more response packets, please note the arrow direction here.
Response packets are sent from server to client only; we can send one or more response packets to the client as a response for its request(s), please note the arrow direction here.
Two Way Packets are sent from server to client or from client to server; we can send one or more two way packets from both sides, please note the arrow direction here.
Event Packets are sent from client to server only; we can send only one Event packet at a time, Event packet is received by the server silently, i.e. there will be no response from the server to the client, please note the arrow direction here.
All generated Packets will have the illustrated OOP Hierarchy ...
Building your TCP Protocol Using .Net Protocol Builder and generating its C# code

Network Programming made easy!

1- Determine the data you want to exchange between clients and the server.
2- Build the packets you decided by adding packets to the suitable node in .Net Protocol Builder (right click the node and select Add Packet, or use Ctrl+A after selecting the suitable node)
3- Add your Fields to the Packets you already added, and change the type of the field according to your design
4- Available Field types are: byte, Int16, Int32, Int64, UInt16, UInt32, UInt64, double, DateTime, string, and byte[] (in last two cases set the maximum expected length, however, the generated protocol will send the actual data only).

Network Programming made easy!

5- when you are done, Generate the code for your protocol by selecting tools-Generate Code or press (Ctrl+G). .Net Protocol Builder will prompt you to select a path where the Protocol files will be saved, **congratulation, your Protocol Code is ready ... !!**
6-The generated files are for the classes illustrated here, using these classes is very easy and straightforward, refer to Server/Client sample code in the help file for more info.

Network Programming made easy!

7-Generate your protocol documentation in docx format by selecting tools-Generate Protocol Documentation (you need to have MS Word 2007 for this option, otherwise use the txt documentation which is generated by default with the protocol code)
Thanks for watching

Protocol Builder
.NET SOLUTIONS

Network Programming made easy!