Subject: A project that "could" be beneficial to the Renegade community.
Posted by Dave Anderson on Tue, 16 Jan 2007 06:22:03 GMT
View Forum Message <> Reply to Message

Topic title: Project: SocketArchive.dll
- -

The Socket Archive is a dynamic link library that makes coding Tcp and Udp connections extremely easy for even the beginner programmer. The dll consists of 2 main classes which are the Tcp and Udp classes.

Usually it takes anywhere over 100 lines of code to program a working connection whether Tcp or Udp. What I have done is condensed this code for the developers to around 2-4 lines of code. The dll itself contains over 500 lines of code, but all the programmer has to do is reference it in their project, code 2 lines of code and everything else is handled for them. This dll is fast in performance and there is hardly any delay in connections. So far the furthest connection was tested from Pueblo, Colorado to the UK and back with less than 2-3 seconds in delay.

As of now, the project is composed of only the Tcp portion in which the only coded part is the TcpListener which listens for incoming connections and handles the TcpClients. The server can accept as many clients as your computer or server can handle and puts each of them on their own thread. This dll is thread safe, so there isn't anything to worry about having cross-thread errors or performance loss issues.

The dll is coded in C++/CLI using .NET 2.0 technology. I have utilized the code to optimize this for optimal performance. The code also maintains C++ ISO Standards.

As of now, the following is what has been done:

Tcp Class
This class is the parent class that houses all Tcp content inside the dll.

Listener Class
The Listener class houses all the methods and handling of the object that handles the incoming client connects and incoming client data. Any connections that are incoming from a remote client are accepted and placed on their own threads to prevent single-thread complications and delays when sending and receiving data. All data is transferred via a network stream object and is converted between bytes and other data as needed.

The listener class only has one constructor in which you do not need to pass any arguements. You simply create and initialize a new instance of this class and the listener handles everything for you. The listener operates by automatically selecting your active network adapters IP Address and listens on Tcp port 80 so there is no port forwarding required.

The custom function that starts the listener has been implemented as StartListener(). (Default: Start() and then Listen().

Other custom implementations will include:

ListenIncomingText();
ListenIncomingBytes();
ListenIncomingFile();
ListenIncomingFiles();

As well as:

ReplyOutgoingText();
ReplyOutgoingBytes();
ReplyOutgoingFile();
ListenIncomingFiles();

Client Class
The Tcp Client has not yet been implemented yet, but will include all functions required to send and receive data as well as connections.

Other advantages of this dll are that all information required to be displayed to the user or application operator is passed to a single console window which will handle all application messages. The choice of a console was to give further optimal performance over that of a graphical user interface design.

The Udp class has yet to be implemented or thought over, but will be very similar to that of the Tcp class.

Please leave any comments/suggestions/flaming/ect. in this thread.

Thank you.
