12/11/2015

Professor: Garth Santor

INFO-5104 - C++ Advanced Topics

C++ Final Project

By: Stephen Mahabir and Kevin Postma

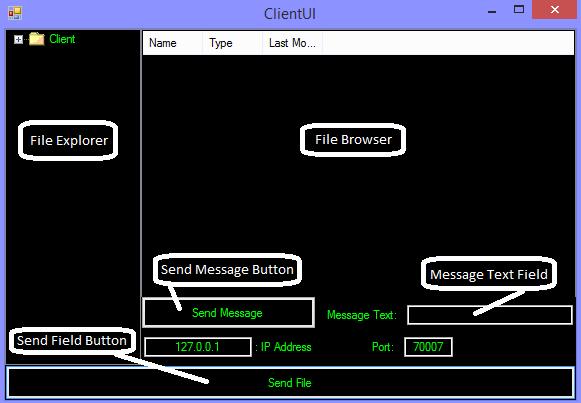
Project 1 – User Guide

# Server



1. Start by selecting the folder in the “File Explorer” to select. This is where all the files will be saved when the Client sends them.
2. You can change the port number or IP Address if you wish before starting to listen.
3. Click “Listener Control Button” this will allow the server to start listening for messages or files being sent.
4. Any Server Status messages (i.e. Receiving a file or a Message) will show on the Server Log
5. Click “Stop Listening” to close all resources and shut down the server

# Client



1. Set the IP and Port number you wish to communicate with.
2. Clients can send Messages to the Server by populating the text box and clicking Send Message.
3. Clients can send Files by browsing to a File using the explorer, selecting a file and clicking “Send File”
4. Clients will get a pop up message of the successfulness of the communication

# Wish List

If we had more time with this application we would have implemented remote file directory searching where a Client could pick where to send the folder to the Server. Another key feature would be handling only a Maximum amount of clients at a time. Having 5 connected clients for example would keep one client in a queue until another client disconnected.

Graphical Interface

Generates .exe

Server UI

* Has one file directory box
* Has one server log box (black background green text
* Will show all server logs and transactions
* Will show all errors and successes
* Will have minimal functionality

Client UI

* Core application!
* Directory browsers
* Buttons for functionality
* Each button will call the wrapper threads (Message or File)
* All messages will be done through message boxes
* Last Error / Success in a textbox at the bottom

DLL Threaded Wrapper

Generates DLL

Server Business Layer

* Runs Listener on a Thread
* Client Calls launch new Thread
* Client Call work is done in the lib
* DLL just ensures work is done
* Reports success / failures to GUI

Client Business layer

* Sender launched on a Thread
* Each Sender Call is done by a separate Thread
* User can make multiple requests
* Each request is done on a separate thread
* Ensures the server checks the queue for more work

Code Library

Generates .lib

Server Class

* Listener
* Receives Server Calls
* Sends / Receives Data
* Has a maximum amount of threads to accept / receive data and ensures wait time is done when to many threads are running at once

Client Class

* Sender
* Calls Server
* Sends Files
* Retrieves Files