

Carl Derline
Karen Bullinger
3-2-2015
CSCE 320
Echo Client Report

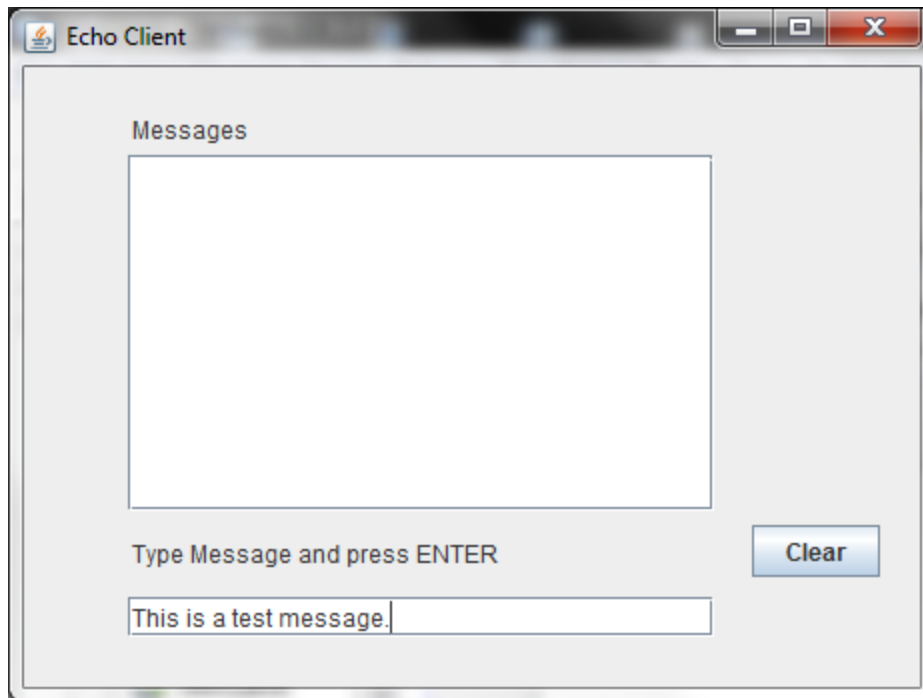
This program is the client portion of the client/server echo application. The echo client sends messages to the server specified on startup and receives messages from the same server. It does this simultaneously by using two threads. One of the threads is created by the `begin()` method (in the controller) and listens exclusively for incoming messages from the server. The other thread, initiated at start up, exclusively sends messages to the server.

In order to get the two threads working simultaneously, we used the `begin()` method to trigger the `run` method on a newly created thread. The `run()` method consists of an infinite loop, continuously waiting for incoming messages from the server. Once a message is received, it is automatically posted to the message text area in the GUI via `updateMessages()`. See screenshot below for an example.

To send messages to the server, the user types a message in the text field and presses enter. The action event inside the view calls `sendMessage()`, which uses the `outputStream` that is connected to the socket to send the message to the server.

The view has one text field, one text area, and one button. The button action event clears the text area's contents. The text field's action event calls `sendMessage()`, which sends the message from the text field to the server. There is also an `updateMessages()` method, which appends the incoming message from the server to the text area.

Message in text area will be sent to server after sending and will also be echoed back into the text area.



Screenshot shows first message sent by user and a second message sent by another user from another computer. Messages were first sent to the server, then echoed back to the client.

