Subject: Socket functions calling order
Posted by captainc on Mon, 14 Jan 2008 17:04:12 GMT
View Forum Message <> Reply to Message

I'm writing a simple client/server socket app and need to confirm the order of operations for the functions and communication with the client. The app will run as a service/daemon and listen for connections from clients.

On the server side,
I first create the ServerSocket

```
if( ServerSocket(sock1, 1555, true, 5, false) ){ ...
```

Then, in a continuous loop using Sleep() function, I check for data on the socket:

```
if( !sock1.IsOpen() ){
    ServerSocket(sock1, 1555, true, 5, false);
}
if( sock1.IsOpen() && !sock1.IsError() && sock1.Peek() && sock1.Accept(sock1) ){
    String sock_data(sock1.Read(2000));
    if(sock_data.GetLength() > 0){
        LOG(String().Cat() << "Data: " << sock_data);
        sock_data.Clear();
        sock1.Clear();
    }
}
```

The client code is simple:

```
if( ClientSocket(sock1, ip_addr, 1555, true, NULL, DEFAULT_CONNECT_TIMEOUT, false) ){
    sock1.Write(String("This is a test"));
    sock1.Close();
}
```

I found that the socket is closed on the server side after calling Read() and then Clear(). If I don't call Clear(), then the program hangs with any subsequent operation on the socket (and the client sees the socket as closed).
If I call Clear(), the socket ends up being closed (as IsOpen() the returns false). I then found that any other operations after Clear() such as Calling IsError() on a closed socket results in an "Assertion failed in ... Core/Other.h, line 17".

Can someone shed some light as to the correctness of the order of the code above?

How would I go about sending a response from the server to the client? Where/when would I call Write() on the socket?