

# Re: socket communication: socket doesn't connect

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.networks/2007-04/msg00280>

---

- *From:* Ananya <[Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 27 Apr 2007 19:50:01 -0700
- 

Thanks! I just had to change the order of things:  
Still first the C++ program brings up the Java program with ShellExecEx.  
Then the C++ program constructs the Server.  
Then the Java program constructs the Client and send some items.  
Then the C++ program accepts and tries to receive the items.

Now the Server constructs and accepts.

However I just accomplished this in the debugger, because I was able to stop the C++ program before it accepts and cause the Java program to first construct the Client and send the items, which in my case has to happen when the Java program gets closed.

I only have two communications between my C++ program and my Java program. To make this clear, let me explain again what my C++ and Java programs do. My Java program draw curves and my C++ program makes it a plugin to Photoshop bringing up its own window. The first communication I need is to bring the image and the workpath from Photoshop into my Java program. And the second communication I need is to bring the curves of my Java program back to Photoshop.

So my final question here is, how can I program my C++ so that after constructing the Server it waits until the Java Client is created and some Java communication has been sent before it tries to accept? Does WSAEventSelect help with that? But how is this done?

Thanks for your patience with me!

"Dave Lowther" wrote:

so why did it not connect?

Your server has to be running for client to be able to connect, but if your server stops running on error 10035 from accept, then it won't be running

Re: socket communication: socket doesn't connect

when you try to connect from your client.

"Ananya" <Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message  
news:BB5AE518-4724-4E8E-822D-BD8B9E060A75@xxxxxxxxxxxxxxxxxxxx

Well, I did call the constructor of the Client in Java:

```
public Client(int p, InetAddress address, int rev) throws IOException,
IllegalArgumentException
{
    port = p;

    try
    {
        sock = new Socket(address, port);
        input = new BufferedInputStream(sock.getInputStream(), BUFFSIZE);
        output = new BufferedOutputStream(sock.getOutputStream(), BUFFSIZE);
    }
    catch (IOException e)
    {
    }
    catch (IllegalArgumentException ie)
    {
    }

    buff = new byte[BUFFSIZE];
    data = new byte[BUFFSIZE];

    output.write(rev);
    output.flush();
}
```

so why did it not connect?

"Dave Lowther" wrote:

10035 = A non-blocking socket operation could not be  
completed  
immediately.

So it's not really an error.

If you want accept() to block until a client connects, then  
don't put the  
socket in non blocking mode.

If you want to poll accept(), then 10035 is to be expected if  
no client  
has

Re: socket communication: socket doesn't connect  
connected since last time you called accept().

HTH

Dave.

"Ananya" <Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote  
in message  
[news:978D94FD-0194-474E-B00C-A85452EF0AE2@xxxxxxxxxxxxxxxxxxxx](mailto:news:978D94FD-0194-474E-B00C-A85452EF0AE2@xxxxxxxxxxxxxxxxxxxx)

Thanks for your response!

Ok, now in my connectServer method I am  
no more calling the connect  
method,  
but I am calling the accept method from  
WinSock2.h. Well, the accept  
method  
also returns -1.

Before calling the accept method,  
WSAGetLastError() returns 0, and  
afterwards it returns 10035.

Please help!

"Michael K. O'Neill" wrote:

"Ananya"  
<Ananya@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>  
wrote in message  
[news:6E597362-E344-4299-84DE-91835941DC4A@xxxxxxxxxxxxxxxxxxxx](mailto:news:6E597362-E344-4299-84DE-91835941DC4A@xxxxxxxxxxxxxxxxxxxx)

I originally  
published  
my question  
with the  
title above  
in the  
Visual  
Tools &  
Languages  
> C/C++ &  
Visual C++  
>  
vc.language  
forum at

Re: socket communication: socket doesn't connect

<http://msdn.microsoft.com/newsgroups/default.aspx?&guid=&sloc=en-us&d>

and I was  
told that it  
really  
belongs  
here. So  
here it is  
again:

I am trying  
to establish  
socket  
communication  
between my  
C++ and  
Java  
program.

I bring up  
the Java  
program  
from the  
C++  
program  
with  
ShellExecEx.

Then I start  
in the Java  
program  
with:  
int port =  
3000;  
int rev = 1;  
InetAddress  
address =  
InetAddress.getLocalHost();  
Client client  
= new  
Client(port,  
address,  
rev);  
int[] ints =  
new int[1];  
ints[0] = 1;  
client.send\_ints(ints,  
1);  
client.closesocket();  
using my

Re: socket communication: socket doesn't connect

Java Client  
class.

Then I  
continue in  
my C++  
program  
with:  
int port =  
3000;  
Server\*  
server =  
new  
Server(port);  
server->connectServer();  
using my  
C++ Server  
class.

It looks like  
the server  
gets  
constructed  
properly  
and the  
connectServer  
method  
calls the  
connect  
method  
from  
WinSock.h,  
which I  
have  
included  
(and  
its library  
WSock32.Lib  
is at  
Additional  
Dependences  
in the Input  
of  
the  
Linker).

Why does  
this connect  
method  
from  
WinSock.h

Re: socket communication: socket doesn't connect

return - 1?

< snip >

To determine why connect() is failing, call WSAGetLastError() and tell us its value. It might be something like WSAENOTSOCK or WSAENOTINITIALISED

Your explanation reverses the "typical" roles of client and server. In a typical scenario, it's the client that calls connect() to connect to the server, whereas in your example, it's the server that's calling connect() to connect to the client. Was that intentional? Frankly, since your C++ Server class is apparently calling bind() and listen(), the Java client should be the one calling Connect(), and your C++ server should be accepting these connections by calling accept().

Mike

Re: socket communication: socket doesn't connect