

# send(), returns Winsock Error 10038

Asked 15 years, 10 months ago   Modified 15 years, 10 months ago   Viewed 5k times



## Problem:

0

- Call to **send()**, returns Winsock Error **10038** against socket handle



## Illustration:



```
acceptedSocket = accept (server, (sockaddr *)&sin, &len);
```

- **accept()**, returns 0
  - A new **thread**, is created for each connection
  - **send()**, (in thread function) returns **10038**

## Illustration: - in thread function

```
//omitted
SOCKET RemoteSocket = (SOCKET) client;
//omitted
send (RemoteSocket, stringToSend, strlen(stringToSend), 0)
```

## Suggestions:

- Possible, race condition?
- Could use I/O completion ports, but not at this stage

c++

windows

visual-studio-2008

multithreading

winsock

Share

Improve this question

Follow

edited Feb 23, 2009 at 15:00

asked Feb 23, 2009 at 14:23



Aaron

2,863 ● 10 ● 44 ● 58

## 3 Answers

Sorted by: Highest score (default)



Isn't the problem in the line

2

```
acceptedSocket = accept (server, (sockaddr *)&sin, &len) == INVALID_SOCKET)
```



You make `acceptedSocket` the result of the comparison, but you should store the actual socket returned from `accept` somehow:



```
acceptedSocket = accept (server, (sockaddr *)&sin, &len);  
isOK= acceptedSocket!=INVALID_SOCKET;
```



Although I'm a bit confused by the unbalanced parentheses in your post, so I may be wrong

[Share](#) [Improve this answer](#) [Follow](#)

answered Feb 23, 2009 at 14:52



[jpalecek](#)

47.7k ● 7 ● 103 ● 146



2



`accept()` returns you a handle to a new connection-specific socket. for server code it's 2+ sockets involved: one is in listen state you are calling `accept()` for and second is one returned from `accept()` - it's an incoming connection socket. Following `accept()` can return socket for second incoming connection etc. if `accept()` returns 0 it's not an incoming connection - it's an error.



[Share](#) [Improve this answer](#) [Follow](#)

answered Feb 23, 2009 at 14:43



[grigoryvp](#)

42.4k ● 66 ● 183 ● 279

One correction: `accept` returning 0 is a normal (successful) execution. Errors are signaled by -1 – [jpalecek](#) Feb 23, 2009 at 14:48



0



Hmm, seems like your `send` is executing too fast before the `accept` happened. So the socket used in `send` is not valid at the point `send` is executed. One of the obnoxious feature of multithreading. You need to wait for an event at the `send` thread and fire an event when an `accept` occurs



[Share](#) [Improve this answer](#) [Follow](#)

answered Feb 23, 2009 at 14:50



[rptony](#)

1,024 ● 2 ● 12 ● 22

A socket handle is defined as "a non-negative integer". So, I believe, zero is valid? – [Aaron](#) Feb 23, 2009 at 16:16