

## Forum: [Hack Forums](#)

Thread: [recv\(\) fails every time..](#)

**F.A.S.T** - July 28, 2018, 11:22 am

I cannot use the Recv function. It always gives me errors.

PHP Code

```
<?php
if ((recv_size = recv(main, server_reply, 2000, 0)) > 0) {
    cout << "Recv failed!" << endl;
    getchar();
    return 1;
}
```

```
server_reply[recv_size] = '\0';
cout << server_reply << endl;
```

```
closesocket(main);
WSACleanup();
getchar();
return 0;
```

Can anyone tell me a fix.

**Auriana** - July 28, 2018, 2:25 pm

You're saying if recv is greater than zero, then it failed. Recv fails on -1, it succeeds if it's above zero.

**sqsgkrldhwg** - July 29, 2018, 2:59 am

Hi,

As said above.

<https://docs.microsoft.com/en-us/windows/desktop/api/winsock/nf-winsock-recv>

Quote: int recv( SOCKET s, char \*buf, int len, int flags);

Return Value

If no error occurs, recv returns the number of bytes received and the buffer pointed to by the buf parameter will contain this data received. If the connection has been gracefully closed, the return value is zero.

Otherwise, a value of SOCKET\_ERROR is returned, and a specific error code can be retrieved by calling WSAGetLastError.

Be aware that sometimes your buffer could be incomplete, I mean the recv function may not completely fill up to len your buffer.

In blocking mode, you can use this function to make sure to receive exactly len bytes.  
If using non blocking mode, this function will be useless ... the implementation is different.

**F.A.S.T** - July 29, 2018, 4:24 am

(Jul 29, 2018 02:59 AM)sqsgkrldhwg Wrote: [\[CITE\]](#) Hi,

As said above.

<https://docs.microsoft.com/en-us/windows/desktop/api/winsock/nf-winsock-recv>

Be aware that sometimes your buffer could be incomplete, I mean the recv function may not completely fill up to len your buffer.

In blocking mode, you can use this function to make sure to receive exactly len bytes.  
If using non blocking mode, this function will be useless ... the implementation is different.  
This explanation was actually good. Thanks a lot for this.  
I'll try this implementation and tell you, Thanks again.

**sqsgkrldhwg** - July 29, 2018, 5:00 am

You seems to be a cool guy.

Bonus shortcuts :

With that you can pass "anything" over the socket.

But it must be used only with basic data (integers and basic arrays). You must not use this to pass objects containing pointers obviously otherwise the pointer will pass over the network itself but the data it refers to.

for instance :

But going in with std::string or anything related will cause serious issues ...  
You must implement a serialize function for that.

Have fun.

**aeiro** - July 29, 2018, 8:13 am

(Jul 29, 2018 05:00 AM)sqsgkrldhwg Wrote: [\[CITE\]](#) But going in with std::string or anything related will cause serious issues ...

You must implement a serialize function for that.

Have fun.

Example of serialization and deserialization with struct of std::string

[https://github.com/HFaeiro/ChatServerAndClient/blob/245db60084b013c163a0605e5b8c5bce38314d6f/Server/HashChatServer/Data\\_Handler.cpp#L23](https://github.com/HFaeiro/ChatServerAndClient/blob/245db60084b013c163a0605e5b8c5bce38314d6f/Server/HashChatServer/Data_Handler.cpp#L23)

struct was defined as:

Code struct \_sData {

```

int Type;
std::string fU;
std::string fP;
std::string tU;
char* Buffer = new char[512];
};

```

**F.A.S.T** - July 29, 2018, 11:25 am

Ok so thanks a lot I actually understood how recv() works.

but I don't know why it returns ASCII trash? Maybe because it's an int? ..

@sqsgkrldwg, Thank you so much, For explaining that to me. I took your reference and did some experiments and the following code Worked..

PHP Code

```

<?php
if (recv_size == recv(main, server_reply, BUFFER, 0) > 0) {
    //server_reply[recv_size] = '\0';
    std::cout << server_reply << std::endl;
}
else {
    std::cout << "recv() returned -1.." << std::endl;
}

```

Also, I do not understand this. Clearly, The if statement is telling that if connect is not SOCKET\_ERROR than raise error? What?

And changing it to == SOCKET\_ERROR it gives an connect Error?

Am I understanding wrong or the code is wrong?

have a look

PHP Code

```

<?php
std::cout << "Attempting to connect to serv " << std::endl;
if (connect(main, (struct sockaddr*)&main, sizeof(main)) != SOCKET_ERROR) {
    std::cout << "Unable to Connect : " << WSAGetLastError();
    getchar();
    return 1;
}
std::cout << "Connected!" << std::endl;

```

**aeiro** - July 30, 2018, 3:55 pm

(Jul 29, 2018 11:25 AM)F.A.S.T Wrote: [\[CITE\]](#) PHP Code

```

<?php
if (connect(main, (struct sockaddr*)&main, sizeof(main)) != SOCKET_ERROR) {
    std::cout << "Unable to Connect : " << WSAGetLastError();
    getchar();
    return 1;
}

```

so what your code is saying is that if it does not = SOCKET\_ERROR then std::cout an error that does not exist.

it should be == SOCKET\_ERROR. then if there is an error it will cout the last error. at that point you should be looking up the error code.

**F.A.S.T** - July 30, 2018, 3:59 pm

(Jul 30, 2018 03:55 PM)aeiro Wrote: [\[CITE\]](#) so what your code is saying is that if it does not = SOCKET\_ERROR then std::cout an error that does not exist.

it should be == SOCKET\_ERROR. then if there is an error it will cout the last error. at that point you should be looking up the error code.

Yeah. I had doubts about it as well. Thanks.