

[<< Prev](#)[Beej's Guide to Network Programming](#)[Next >>](#)

1. Intro

Hey! Socket programming got you down? Is this stuff just a little too difficult to figure out from the **man** pages? You want to do cool Internet programming, but you don't have time to wade through a gob of `structs` trying to figure out if you have to call `bind()` before you `connect()`, etc., etc.

Well, guess what! I've already done this nasty business, and I'm dying to share the information with everyone! You've come to the right place. This document should give the average competent C programmer the edge s/he needs to get a grip on this networking noise.

And check it out: I've finally caught up with the future (just in the nick of time, too!) and have updated the Guide for IPv6! Enjoy!

1.1. Audience

This document has been written as a tutorial, not a complete reference. It is probably at its best when read by individuals who are just starting out with socket programming and are looking for a foothold. It is certainly not the *complete and total* guide to sockets programming, by any means.

Hopefully, though, it'll be just enough for those man pages to start making sense... :-)

1.2. Platform and Compiler

The code contained within this document was compiled on a Linux PC using Gnu's `gcc` compiler. It should, however, build on just about any platform that uses `gcc`. Naturally, this doesn't apply if you're programming for Windows—see the [section on Windows programming](#), below.

1.3. Official Homepage and Books For Sale

This official location of this document is <http://beej.us/guide/bgnet/>. There you will also find example code and translations of the guide into various languages.

To buy nicely bound print copies (some call them "books"), visit <http://beej.us/guide/url/bgbuy>. I'll appreciate the purchase because it helps sustain my document-writing lifestyle!

1.4. Note for Solaris/SunOS Programmers

When compiling for Solaris or SunOS, you need to specify some extra command-line switches for linking in the proper libraries. In order to do this, simply add "`-lnsl -lsocket -lresolv`" to the end of the compile command, like so:

```
$ cc -o server server.c -lnsl -lsocket -lresolv
```

If you still get errors, you could try further adding a "-lXnet" to the end of that command line. I don't know what that does, exactly, but some people seem to need it.

Another place that you might find problems is in the call to `setsockopt()`. The prototype differs from that on my Linux box, so instead of:

```
int yes=1;
```

enter this:

```
char yes='1';
```

As I don't have a Sun box, I haven't tested any of the above information—it's just what people have told me through email.

1.5. Note for Windows Programmers

At this point in the guide, historically, I've done a bit of bagging on Windows, simply due to the fact that I don't like it very much. But I should really be fair and tell you that Windows has a huge install base and is obviously a perfectly fine operating system.

They say absence makes the heart grow fonder, and in this case, I believe it to be true. (Or maybe it's age.) But what I can say is that after a decade-plus of not using Microsoft OSes for my personal work, I'm much happier! As such, I can sit back and safely say, "Sure, feel free to use Windows!" ...Ok yes, it does make me grit my teeth to say that.

So I still encourage you to try [Linux](#), [BSD](#), or some flavor of Unix, instead.

But people like what they like, and you Windows folk will be pleased to know that this information is generally applicable to you guys, with a few minor changes, if any.

One cool thing you can do is install [Cygwin](#), which is a collection of Unix tools for Windows. I've heard on the grapevine that doing so allows all these programs to compile unmodified.

But some of you might want to do things the Pure Windows Way. That's very gutsy of you, and this is what you have to do: run out and get Unix immediately! No, no—I'm kidding. I'm supposed to be Windows-friendly(er) these days...

This is what you'll have to do (unless you install [Cygwin](#)!): first, ignore pretty much all of the system header files I mention in here. All you need to include is:

```
#include <winsock.h>
```

Wait! You also have to make a call to `WSAStartup()` before doing anything else with the sockets library. The code to do that looks something like this:

```
#include <winsock.h>
{
    WSADATA wsaData;    // if this doesn't work
```

```
//WSAData wsaData; // then try this instead

// MAKEWORD(1,1) for Winsock 1.1, MAKEWORD(2,0) for Winsock 2.0:

if (WSAStartup(MAKEWORD(1,1), &wsaData) != 0) {
    fprintf(stderr, "WSAStartup failed.\n");
    exit(1);
}
```

You also have to tell your compiler to link in the Winsock library, usually called *wsock32.lib* or *winsock32.lib*, or *ws2_32.lib* for Winsock 2.0. Under VC++, this can be done through the Project menu, under Settings... Click the Link tab, and look for the box titled "Object/library modules". Add "wsock32.lib" (or whichever lib is your preference) to that list.

Or so I hear.

Finally, you need to call **WSACleanup()** when you're all through with the sockets library. See your online help for details.

Once you do that, the rest of the examples in this tutorial should generally apply, with a few exceptions. For one thing, you can't use **close()** to close a socket—you need to use **closesocket()**, instead. Also, **select()** only works with socket descriptors, not file descriptors (like 0 for `stdin`).

There is also a socket class that you can use, `CSocket`. Check your compilers help pages for more information.

To get more information about Winsock, read the [Winsock FAQ](#) and go from there.

Finally, I hear that Windows has no **fork()** system call which is, unfortunately, used in some of my examples. Maybe you have to link in a POSIX library or something to get it to work, or you can use **CreateProcess()** instead. **fork()** takes no arguments, and **CreateProcess()** takes about 48 billion arguments. If you're not up to that, the **CreateThread()** is a little easier to digest...unfortunately a discussion about multithreading is beyond the scope of this document. I can only talk about so much, you know!

1.6. Email Policy

I'm generally available to help out with email questions so feel free to write in, but I can't guarantee a response. I lead a pretty busy life and there are times when I just can't answer a question you have. When that's the case, I usually just delete the message. It's nothing personal; I just won't ever have the time to give the detailed answer you require.

As a rule, the more complex the question, the less likely I am to respond. If you can narrow down your question before mailing it and be sure to include any pertinent information (like platform, compiler, error messages you're getting, and anything else you think might help me troubleshoot), you're much more likely to get a response. For more pointers, read ESR's document, [How To Ask Questions The Smart Way](#).

If you don't get a response, hack on it some more, try to find the answer, and if it's still elusive, then write me again with the information you've found and hopefully it will be enough for me to help out.

Now that I've badgered you about how to write and not write me, I'd just like to let you know that I *fully* appreciate all the praise the guide has received over the years. It's a real morale boost, and it gladdens me to hear that it is being used for good! :-) Thank you!

1.7. Mirroring

You are more than welcome to mirror this site, whether publicly or privately. If you publicly mirror the site and want me to link to it from the main page, drop me a line at beej@beej.us.

1.8. Note for Translators

If you want to translate the guide into another language, write me at beej@beej.us and I'll link to your translation from the main page. Feel free to add your name and contact info to the translation.

Please note the license restrictions in the Copyright and Distribution section, below.

If you want me to host the translation, just ask. I'll also link to it if you want to host it; either way is fine.

1.9. Copyright and Distribution

Beej's Guide to Network Programming is Copyright © 2012 Brian "Beej Jorgensen" Hall.

With specific exceptions for source code and translations, below, this work is licensed under the Creative Commons Attribution- Noncommercial- No Derivative Works 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

One specific exception to the "No Derivative Works" portion of the license is as follows: this guide may be freely translated into any language, provided the translation is accurate, and the guide is reprinted in its entirety. The same license restrictions apply to the translation as to the original guide. The translation may also include the name and contact information for the translator.

The C source code presented in this document is hereby granted to the public domain, and is completely free of any license restriction.

Educators are freely encouraged to recommend or supply copies of this guide to their students.

Contact beej@beej.us for more information.

[<< Prev](#)

[Beej's Guide to Network Programming](#)

[Next >>](#)