



# Convert an IP address to the corresponding domain name in C

---

## Content

- [1 Objective](#)
- [2 Scenario](#)
- [3 Method](#)
- [4 See also](#)
- [5 Further reading](#)

## Objective

To convert an IPv4 or IPv6 address to the corresponding domain name (if there is one) in C

## Tested on

Debian (Lenny, Squeeze)  
Ubuntu (Lucid, Precise, Trusty)

## Scenario

Suppose you have used the `getpeername` function to obtain the remote address to which a particular TCP socket is connected:

```
struct sockaddr_storage addr;
socklen_t addr_len=sizeof(addr);
int err=getpeername(sock_fd, (struct sockaddr*)&addr, &addr_len);
if (err!=0) {
    die("failed to fetch remote address (errno=%d)", errno);
}
```

The remote address has been written to a buffer called `addr`. This buffer is of type `struct sockaddr_storage`, but the address stored within it will be of type `struct sockaddr_in` or `struct sockaddr_in6`. The length of the address has been recorded in the variable `addr_len`. Note that:

- `addr` is a socket address, so in addition to the IP address it contains information such as the address family and port number.
- `addr_len` will probably not be equal to `sizeof(struct sockaddr_storage)` once the call to `getpeername` has completed.

You wish to convert the IP address contained within `addr` to the corresponding domain name (if there is one), or failing that to a human-readable numeric representation.

## Method

One way to perform the required conversion is to call the `getnameinfo` function:

```
#include <netdb.h>
#include <sys/socket.h>
#include <netinet/in.h>

#ifdef NI_MAXHOST
#define NI_MAXHOST 1025
#endif

// ...

char buffer[NI_MAXHOST];
int err=getnameinfo((struct sockaddr*)&addr,addr_len,buffer,sizeof(buffer),0,0,0);
if (err!=0) {
    die("failed to convert address to string (code=%d)",err);
}
printf("Remote address: %s\n",buffer);
```

If the IP address cannot be resolved to a hostname then it will instead be converted to a human-readable numeric format (for example 192.168.0.1 or 2001:db8::1).

POSIX does not define any fixed upper limit on the size of buffer that might be needed, however implementations often provide a macro called `NI_MAXHOST` for this purpose. For portability you should test whether this macro exists and provide your own value if it does not. The value used by glibc at the time of writing was 1025.

## See also

- [Convert an IP address to the corresponding domain name](#)
- [Convert an IP address to a human-readable string in C](#)

## Further reading

- [getnameinfo](#), *The Open Group Base Specifications*, Issue 7, The Open Group, 2008

Tags: [c](#) | [dns](#) | [posix](#)

© 2010–2014 [Graham Shaw](#), some rights reserved.