

**NAME**

getnetent, getnetbyname, getnetbyaddr, setnetent, endnetent – get network entry

**SYNOPSIS**

```
#include <netdb.h>

struct netent *getnetent(void);
struct netent *getnetbyname(const char *name);
struct netent *getnetbyaddr(uint32_t net, int type);
void setnetent(int stayopen);
void endnetent(void);
```

**DESCRIPTION**

The **getnetent()** function reads the next entry from the networks database and returns a *netent* structure containing the broken-out fields from the entry. A connection is opened to the database if necessary.

The **getnetbyname()** function returns a *netent* structure for the entry from the database that matches the network *name*.

The **getnetbyaddr()** function returns a *netent* structure for the entry from the database that matches the network number *net* of type *type*. The *net* argument must be in host byte order.

The **setnetent()** function opens a connection to the database, and sets the next entry to the first entry. If *stayopen* is nonzero, then the connection to the database will not be closed between calls to one of the **getnet\*()** functions.

The **endnetent()** function closes the connection to the database.

The *netent* structure is defined in *<netdb.h>* as follows:

```
struct netent {
    char      *n_name;      /* official network name */
    char      **n_aliases; /* alias list */
    int       n_addrtype; /* net address type */
    uint32_t  n_net;       /* network number */
}
```

The members of the *netent* structure are:

*n\_name*

The official name of the network.

*n\_aliases*

A NULL-terminated list of alternative names for the network.

*n\_addrtype*

The type of the network number; always **AF\_INET**.

*n\_net*

The network number in host byte order.

**RETURN VALUE**

The **getnetent()**, **getnetbyname()** and **getnetbyaddr()** functions return a pointer to a statically allocated *netent* structure, or a null pointer if an error occurs or the end of the file is reached.

**FILES**

*/etc/networks*

networks database file

**ATTRIBUTES**

For an explanation of the terms used in this section, see [attributes\(7\)](#).

Interface	Attribute	Value
<b>getnetent()</b>	Thread safety	MT-Unsafe race:netent race:netentbuf env locale
<b>getnetbyname()</b>	Thread safety	MT-Unsafe race:netbyname env locale
<b>getnetbyaddr()</b>	Thread safety	MT-Unsafe race:netbyaddr locale
<b>setnetent(), endnetent()</b>	Thread safety	MT-Unsafe race:netent env locale

In the above table, *netent* in *race:netent* signifies that if any of the functions **setnetent()**, **getnetent()**, or **endnetent()** are used in parallel in different threads of a program, then data races could occur.

### CONFORMING TO

POSIX.1-2001, POSIX.1-2008, 4.3BSD.

### NOTES

In glibc versions before 2.2, the *net* argument of **getnetbyaddr()** was of type *long*.

### SEE ALSO

[getnetent\\_r\(3\)](#), [getprotoent\(3\)](#), [getservent\(3\)](#)  
RFC 1101

### COLOPHON

This page is part of release 4.16 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.