

**NAME**

getnetent\_r, getnetbyname\_r, getnetbyaddr\_r – get network entry (reentrant)

**LIBRARY**

Standard C library (*libc*, *-lc*)

**SYNOPSIS**

```
#include <netdb.h>
```

```
int getnetent_r(struct netent *restrict result_buf,
               char buf[restrict .buflen], size_t buflen,
               struct netent **restrict result,
               int *restrict h_errnop);
int getnetbyname_r(const char *restrict name,
                  struct netent *restrict result_buf,
                  char buf[restrict .buflen], size_t buflen,
                  struct netent **restrict result,
                  int *restrict h_errnop);
int getnetbyaddr_r(uint32_t net, int type,
                  struct netent *restrict result_buf,
                  char buf[restrict .buflen], size_t buflen,
                  struct netent **restrict result,
                  int *restrict h_errnop);
```

Feature Test Macro Requirements for glibc (see **feature\_test\_macros(7)**):

**getnetent\_r()**, **getnetbyname\_r()**, **getnetbyaddr\_r()**:

Since glibc 2.19:

`_DEFAULT_SOURCE`

glibc 2.19 and earlier:

`_BSD_SOURCE` || `_SVID_SOURCE`

**DESCRIPTION**

The **getnetent\_r()**, **getnetbyname\_r()**, and **getnetbyaddr\_r()** functions are the reentrant equivalents of, respectively, **getnetent(3)**, **getnetbyname(3)**, and **getnetbynumber(3)**. They differ in the way that the *netent* structure is returned, and in the function calling signature and return value. This manual page describes just the differences from the nonreentrant functions.

Instead of returning a pointer to a statically allocated *netent* structure as the function result, these functions copy the structure into the location pointed to by *result\_buf*.

The *buf* array is used to store the string fields pointed to by the returned *netent* structure. (The nonreentrant functions allocate these strings in static storage.) The size of this array is specified in *buflen*. If *buf* is too small, the call fails with the error **ERANGE**, and the caller must try again with a larger buffer. (A buffer of length 1024 bytes should be sufficient for most applications.)

If the function call successfully obtains a network record, then *\*result* is set pointing to *result\_buf*; otherwise, *\*result* is set to **NULL**.

The buffer pointed to by *h\_errnop* is used to return the value that would be stored in the global variable *h\_errno* by the nonreentrant versions of these functions.

**RETURN VALUE**

On success, these functions return 0. On error, they return one of the positive error numbers listed in **ERRORS**.

On error, record not found (**getnetbyname\_r()**, **getnetbyaddr\_r()**), or end of input (**getnetent\_r()**) *result* is set to **NULL**.

**ERRORS**

**ENOENT**

(**getnetent\_r**()) No more records in database.

**ERANGE**

*buf* is too small. Try again with a larger buffer (and increased *buflen*).

**ATTRIBUTES**

For an explanation of the terms used in this section, see **attributes**(7).

Interface	Attribute	Value
<b>getnetent_r</b> (), <b>getnetbyname_r</b> (), <b>getnetbyaddr_r</b> ()	Thread safety	MT-Safe locale

**STANDARDS**

These functions are GNU extensions. Functions with similar names exist on some other systems, though typically with different calling signatures.

**SEE ALSO**

**getnetent**(3), **networks**(5)