



## ***Rocky Enterprise Linux 9.2 Manual Pages on command 'getnetbyaddr\_r.3'***

**C:\>man getnetbyaddr\_r.3**

GETNETENT\_R(3)                   Linux Programmer's Manual                   GETNETENT\_R(3)

### NAME

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

### SYNOPSIS

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

```
int getnetent_r(struct netent *result_buf, char *buf,  
               size_t buflen, struct netent **result,  
               int *h_errnop);
```

```
int getnetbyname_r(const char *name,  
                  struct netent *result_buf, char *buf,  
                  size_t buflen, struct netent **result,  
                  int *h_errnop);
```

```
int getnetbyaddr_r(uint32_t net, int type,  
                  struct netent *result_buf, char *buf,  
                  size_t buflen, struct netent **result,  
                  int *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 `getnetbyaddr(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 ?

??

?`getnetent_r()`, ? Thread safety ? MT-Safe locale ?

?`getnetbyname_r()`, ? ? ?

