



Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'getnetbyaddr_r.3'

\$ man getnetbyaddr_r.3

getnetent_r(3) Library Functions Manual getnetent_r(3)

NAME

getnetent_r, getnetbyname_r, getnetbyaddr_r - get network entry (reentrant)

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

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

```
int getnetent_r(size_t size;
```

```
    struct netent *restrict result_buf,
```

```
    char buf[restrict size], size_t size,
```

```
    struct netent **restrict result,
```

```
    int *restrict h_errnop);
```

```
int getnetbyname_r(size_t size;
```

```
    const char *restrict name,
```

```
    struct netent *restrict result_buf,
```

```
    char buf[restrict size], size_t size,
```

```
    struct netent **restrict result,
```

```
    int *restrict h_errnop);
```

```
int getnetbyaddr_r(size_t size;
```

```
    uint32_t net, int type,
```

```
    struct netent *restrict result_buf,
```

```
    char buf[restrict size], size_t size,
```

```
    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 `size`. 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 size 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 size).

ATTRIBUTES

```

? Interface      ? Attribute   ? Value        ?
? getnetent_r(), getnetbyname_r(), getnetbyaddr_r()    ? Thread safety ? MT-Safe locale ?
```

Functions with similar names exist on some other systems, though typically with different calling signatures.

GNU.

```
getnetent(3), networks(5)
```

Page 3/3