



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

\$ man endnetgrent.3

setnetgrent(3) Library Functions Manual setnetgrent(3)

NAME

setnetgrent, endnetgrent, getnetgrent, getnetgrent_r, innetgr - handle network group entries

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

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

```
int setnetgrent(const char *netgroup);
```

```
void endnetgrent(void);
```

```
int getnetgrent(char **restrict host,  
                char **restrict user, char **restrict domain);
```

```
int getnetgrent_r(size_t size;  
                  char **restrict host,  
                  char **restrict user, char **restrict domain,  
                  char buff[restrict size], size_t size);
```

```
int innetgr(const char *netgroup, const char *host,  
            const char *user, const char *domain);
```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

setnetgrent(), endnetgrent(), getnetgrent(), getnetgrent_r(), innetgr():

Since glibc 2.19:

```
_DEFAULT_SOURCE
```

glibc 2.19 and earlier:

```
_BSD_SOURCE || _SVID_SOURCE
```

DESCRIPTION

The `netgroup` is a SunOS invention. A `netgroup` database is a list of string triples (host? name, username, domainname) or other `netgroup` names. Any of the elements in a triple can be empty, which means that anything matches. The functions described here allow access to the `netgroup` databases. The file `/etc/nsswitch.conf` defines what database is searched.

The `setnetgrent()` call defines the netgroup that will be searched by subsequent `getnetgrent()` calls. The `getnetgrent()` function retrieves the next netgroup entry, and returns pointers in `host`, `user`, `domain`. A null pointer means that the corresponding entry matches any string. The pointers are valid only as long as there is no call to other netgroup-related functions. To avoid this problem you can use the GNU function `getnetgrent_r()` that stores the strings in the supplied buffer. To free all allocated buffers use `endnetgrent()`.

In most cases you want to check only if the triplet (hostname, username, domainname) is a member of a netgroup. The function `innetgr()` can be used for this without calling the above three functions. Again, a null pointer is a wildcard and matches any string. The function is thread-safe.

RETURN VALUE

These functions return 1 on success and 0 for failure.

FILES

```
/etc/netgroup
```

```
/etc/nsswitch.conf
```

ATTRIBUTES

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

[illegible]

?	Interface	?	Attribute	?	Value	?
---	-----------	---	-----------	---	-------	---

[illegible]

? setnetgrent(), ? Thread safety ? MT-Unsafe race:netgrent locale ?

? getnetgrent_r(), ? ?

? innergr() ? ? ?

[illegible]

? endnetgrent() ? Thread safety ? MT-Unsafe race:netgrent ?

[illegible]

? getnetgrent() ? Thread safety ? MT-Unsafe race:netgrent race:netgrentbuf locale ?

Page 2/3

In the above table, `netgrent` in `race:netgrent` signifies that if any of the functions `setnetgrent()`, `getnetgrent_r()`, `innetgr()`, `getnetgrent()`, or `endnetgrent()` are used in parallel in different threads of a program, then data races could occur.

VERSIONS

In the BSD implementation, `setnetgrent()` returns `void`.

STANDARDS

None.

HISTORY

`setnetgrent()`, `endnetgrent()`, `getnetgrent()`, and `innetgr()` are available on most UNIX systems. `getnetgrent_r()` is not widely available on other systems.

SEE ALSO

`sethostent(3)`, `setprotoent(3)`, `setservent(3)`

Linux man-pages 6.17

2026-02-08

`setnetgrent(3)`