NAME

getnetconfig, **setnetconfig**, **endnetconfig**, **getnetconfigent**, **freenetconfigent**, **nc_perror**, **nc_sperror** - get network configuration database entry

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <netconfig.h>

*struct netconfig **
getnetconfig(*void *handlep*);

*void **
setnetconfig(*void*);

int
endnetconfig(*void *handlep*);

*struct netconfig **
getnetconfigent(*const char *netid*);

void
freenetconfigent(*struct netconfig *netconfigp*);

void
nc_perror(*const char *msg*);

*char **
nc_sperror(*void*);

DESCRIPTION

The library routines described on this page provide the application access to the system network configuration database, */etc/netconfig*. The **getnetconfig**() function returns a pointer to the current entry in the netconfig database, formatted as a *struct netconfig*. Successive calls will return successive netconfig entries in the netconfig database. The **getnetconfig**() function can be used to search the entire netconfig file. The **getnetconfig**() function returns NULL at the end of the file. The *handlep* argument is the handle obtained through **setnetconfig**().

A call to **setnetconfig**() has the effect of "binding" to or "rewinding" the netconfig database. The

setnetconfig() function must be called before the first call to **getnetconfig**() and may be called at any other time. The **setnetconfig**() function need not be called before a call to **getnetconfigent**(). The **setnetconfig**() function returns a unique handle to be used by **getnetconfig**().

The **endnetconfig**() function should be called when processing is complete to release resources for reuse. The *handlep* argument is the handle obtained through **setnetconfig**(). Programmers should be aware, however, that the last call to **endnetconfig**() frees all memory allocated by **getnetconfig**() for the *struct netconfig* data structure. The **endnetconfig**() function may not be called before **setnetconfig**().

The **getnetconfigent**() function returns a pointer to the netconfig structure corresponding to *netid*. It returns NULL if *netid* is invalid (that is, does not name an entry in the netconfig database).

The **freenetconfigent**() function frees the netconfig structure pointed to by *netconfigp* (previously returned by **getnetconfigent**()).

The **nc_perror**() function prints a message to the standard error indicating why any of the above routines failed. The message is prepended with the string *msg* and a colon. A newline character is appended at the end of the message.

The **nc_sperror**() function is similar to **nc_perror**() but instead of sending the message to the standard error, will return a pointer to a string that contains the error message.

The **nc_perror**() and **nc_sperror**() functions can also be used with the NETPATH access routines defined in getnetpath(3).

RETURN VALUES

The **setnetconfig**() function returns a unique handle to be used by **getnetconfig**(). In the case of an error, **setnetconfig**() returns NULL and **nc_perror**() or **nc_sperror**() can be used to print the reason for failure.

The **getnetconfig**() function returns a pointer to the current entry in the netconfig database, formatted as a *struct netconfig*. The **getnetconfig**() function returns NULL at the end of the file, or upon failure.

The **endnetconfig**() function returns 0 on success and -1 on failure (for example, if **setnetconfig**() was not called previously).

On success, **getnetconfigent**() returns a pointer to the *struct netconfig* structure corresponding to *netid*; otherwise it returns NULL.

The **nc_sperror**() function returns a pointer to a buffer which contains the error message string. This buffer is overwritten on each call. In multithreaded applications, this buffer is implemented as thread-

specific data.

FILES

/etc/netconfig

SEE ALSO

getnetpath(3), netconfig(5)