NAME

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

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);

MT-LEVEL

MT-Safe

DESCRIPTION

The library routines described on this page are part of the Network Selection component. They provide the application access to the system network configuration database, **/etc/netconfig**. In addition to the routines for accessing the **netconfig** database, Network Selection includes the environment variable **NETPATH** (see **environ**(5)) and the **NETPATH** access routines described in **getnetpath**(3N).

getnetconfig( ) 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. **getnetconfig( )** can be used to search the entire **netconfig** file. **getnetconfig( )** returns NULL at the end of the file. *handlep* is the handle obtained through **setnetconfig( )**.

A call to **setnetconfig( )** has the effect of ''binding'' to or ''rewinding'' the **netconfig** database. **setnetconfig( )** must be called before the first call to **getnetconfig( )** and may be called at any other time. **setnetconfig( )** need *not* be called before a call to **getnetconfigent( )**. **setnetconfig( )** returns a unique handle to be used by **getnetconfig( ) .**

endnetconfig( ) should be called when processing is complete to release resources for reuse. *handlep* 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. **endnetconfig( )** may not be called before **setnetconfig( )**.

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

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

nc_perror( ) 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 is appended at the end of the message.

nc_sperror( ) 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.

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

RETURN VALUES

setnetconfig( ) 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.

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

endnetconfig( ) 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.

nc_sperror( ) 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.

SEE ALSO

getnetpath(3N), **netconfig**(4), **environ**(5)

ONC+ Developers Guide
Transport Interfaces Programming Guide