

**NAME**

**getnetpath**, **setnetpath**, **endnetpath** - get */etc/netconfig* entry corresponding to NETPATH component

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <netconfig.h>
```

```
struct netconfig *
```

```
getnetpath(void *handlep);
```

```
void *
```

```
setnetpath(void);
```

```
int
```

```
endnetpath(void *handlep);
```

**DESCRIPTION**

The routines described in this page provide the application access to the system network configuration database, */etc/netconfig*, as it is "filtered" by the NETPATH environment variable (see [environ\(7\)](#)). See [getnetconfig\(3\)](#) for other routines that also access the network configuration database directly. The NETPATH variable is a list of colon-separated network identifiers.

The **getnetpath()** function returns a pointer to the netconfig database entry corresponding to the first valid NETPATH component. The netconfig entry is formatted as a *struct netconfig*. On each subsequent call, **getnetpath()** returns a pointer to the netconfig entry that corresponds to the next valid NETPATH component. The **getnetpath()** function can thus be used to search the netconfig database for all networks included in the NETPATH variable. When NETPATH has been exhausted, **getnetpath()** returns NULL.

A call to **setnetpath()** "binds" to or "rewinds" NETPATH. The **setnetpath()** function must be called before the first call to **getnetpath()** and may be called at any other time. It returns a handle that is used by **getnetpath()**.

The **getnetpath()** function silently ignores invalid NETPATH components. A NETPATH component is invalid if there is no corresponding entry in the netconfig database.

If the NETPATH variable is unset, **getnetpath()** behaves as if NETPATH were set to the sequence of "default" or "visible" networks in the netconfig database, in the order in which they are listed.

The **endnetpath()** function may be called to "unbind" from NETPATH when processing is complete, releasing resources for reuse. Programmers should be aware, however, that **endnetpath()** frees all memory allocated by **getnetpath()** for the struct netconfig data structure.

## RETURN VALUES

The **setnetpath()** function returns a handle that is used by **getnetpath()**. In case of an error, **setnetpath()** returns NULL.

The **endnetpath()** function returns 0 on success and -1 on failure (for example, if **setnetpath()** was not called previously). The **nc\_perror()** or **nc\_spperror()** function can be used to print out the reason for failure. See [getnetconfig\(3\)](#).

When first called, **getnetpath()** returns a pointer to the netconfig database entry corresponding to the first valid NETPATH component. When NETPATH has been exhausted, **getnetpath()** returns NULL.

## SEE ALSO

[getnetconfig\(3\)](#), [netconfig\(5\)](#), [environ\(7\)](#)