NAME

ethers, **ether_line**, **ether_aton**, **ether_aton_r**, **ether_ntoa**, **ether_ntoa_r**, **ether_ntohost**, **ether_hostton** - Ethernet address conversion and lookup routines

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/types.h>
#include <sys/socket.h>
#include <net/ethernet.h>

int
ether_line(*const char *l, struct ether_addr *e, char *hostname*);

*struct ether_addr **
ether_aton(*const char *a*);

*struct ether_addr **
ether_aton_r(*const char *a, struct ether_addr *e*);

*char **
ether_ntoa(*const struct ether_addr *n*);

*char **
ether_ntoa_r(*const struct ether_addr *n, char *buf*);

int
ether_ntohost(*char *hostname, const struct ether_addr *e*);

int
ether_hostton(*const char *hostname, struct ether_addr *e*);

DESCRIPTION

These functions operate on ethernet addresses using an *ether_addr* structure, which is defined in the header file *<net/ethernet.h>*:

```
/*
 * The number of bytes in an ethernet (MAC) address.
 */
```

```
#define ETHER_ADDR_LEN          6

/*
 * Structure of a 48-bit Ethernet address.
 */
struct  ether_addr {
        u_char octet[ETHER_ADDR_LEN];
};
```

The function **ether_line**() scans *l*, an ASCII string in ethers(5) format and sets *e* to the ethernet address specified in the string and *h* to the hostname. This function is used to parse lines from */etc/ethers* into their component parts.

The **ether_aton**() and **ether_aton_r**() functions convert ASCII representation of ethernet addresses into *ether_addr* structures. Likewise, the **ether_ntoa**() and **ether_ntoa_r**() functions convert ethernet addresses specified as *ether_addr* structures into ASCII strings.

The **ether_ntohost**() and **ether_hostton**() functions map ethernet addresses to their corresponding hostnames as specified in the */etc/ethers* database. The **ether_ntohost**() function converts from ethernet address to hostname, and **ether_hostton**() converts from hostname to ethernet address.

RETURN VALUES

The **ether_line**() function returns zero on success and non-zero if it was unable to parse any part of the supplied line *l*. It returns the extracted ethernet address in the supplied *ether_addr* structure *e* and the hostname in the supplied string *h*.

On success, **ether_ntoa**() and **ether_ntoa_r**() functions return a pointer to a string containing an ASCII representation of an ethernet address. If it is unable to convert the supplied *ether_addr* structure, it returns a NULL pointer. **ether_ntoa**() stores the result in a static buffer; **ether_ntoa_r**() stores the result in a user-passed buffer.

Likewise, **ether_aton**() and **ether_aton_r**() return a pointer to an *ether_addr* structure on success and a NULL pointer on failure. **ether_aton**() stores the result in a static buffer; **ether_aton_r**() stores the result in a user-passed buffer.

The **ether_ntohost**() and **ether_hostton**() functions both return zero on success or non-zero if they were unable to find a match in the */etc/ethers* database.

NOTES

The user must ensure that the hostname strings passed to the **ether_line**(), **ether_ntohost**() and

ether_hostton() functions are large enough to contain the returned hostnames.

NIS INTERACTION

If the */etc/ethers* contains a line with a single + in it, the **ether_ntohost**() and **ether_hostton**() functions will attempt to consult the NIS *ethers.byname* and *ethers.byaddr* maps in addition to the data in the */etc/ethers* file.

SEE ALSO

ethers(5), yp(8)

HISTORY

This particular implementation of the **ethers** library functions were written for and first appeared in FreeBSD 2.1. Thread-safe function variants first appeared in FreeBSD 7.0.

BUGS

The **ether_aton**() and **ether_ntoa**() functions returns values that are stored in static memory areas which may be overwritten the next time they are called.

ether_ntoa_r() accepts a character buffer pointer, but not a buffer length. The caller must ensure adequate space is available in the buffer in order to avoid a buffer overflow.