
*Full credit is given to the above companies including the OS that this PDF file was generated!*

## *Red Hat Enterprise Linux Release 9.2 Manual Pages on 'netdb.h.0p' command*

***$ man netdb.h.0p***

netdb.h(0P) POSIX Programmer's Manual netdb.h(0P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

netdb.h ? definitions for network database operations

SYNOPSIS

#include <netdb.h>

DESCRIPTION

The <netdb.h> header may define the in_port_t type and the in_addr_t type as described in <netinet/in.h>.

The <netdb.h> header shall define the hostent structure, which shall include at least the following members:

char *h_name Official name of the host.

char **h_aliases A pointer to an array of pointers to alternative host names, terminated by a null pointer.

int h_addrtype Address type.

int h_length The length, in bytes, of the address.

char **h_addr_list A pointer to an array of pointers to network addresses (in network byte order) for the host,

terminated by a null pointer.

The <netdb.h> header shall define the netent structure, which shall in?

clude at least the following members:

char *n_name Official, fully-qualified (including the domain) name of the host.

char **n_aliases A pointer to an array of pointers to alternative network names, terminated by a null pointer.

int n_addrtype The address type of the network.

uint32_t n_net The network number, in host byte order.

The <netdb.h> header shall define the uint32_t type as described in <inttypes.h>.

The <netdb.h> header shall define the protoent structure, which shall include at least the following members:

char *p_name Official name of the protocol.

char **p_aliases A pointer to an array of pointers to alternative protocol names, terminated by a null pointer.

int p_proto The protocol number.

The <netdb.h> header shall define the servent structure, which shall include at least the following members:

char *s_name Official name of the service.

char **s_aliases A pointer to an array of pointers to alternative service names, terminated by a null pointer.

int s_port A value which, when converted to uint16_t, yields the port number in network byte order at which the service resides.

char *s_proto The name of the protocol to use when contacting the service.

The <netdb.h> header shall define the IPPORT_RESERVED symbolic constant with the value of the highest reserved Internet port number.

The <netdb.h> header shall define the addrinfo structure, which shall include at least the following members:

```
int             ai_flags      Input flags.
int             ai_family     Address family of socket.
int             ai_socktype   Socket type.
int             ai_protocol   Protocol of socket.
socklen_t       ai_addrlen    Length of socket address.
struct sockaddr *ai_addr      Socket address of socket.
char            *ai_canonname Canonical name of service location.
struct addrinfo *ai_next      Pointer to next in list.
```

The <netdb.h> header shall define the following symbolic constants that evaluate to bitwise-distinct integer constants for use in the flags field of the addrinfo structure:

AI_PASSIVE Socket address is intended for bind().

AI_CANONNAME Request for canonical name.

AI_NUMERICHOST

Return numeric host address as name.

AI_NUMERICSERV

Inhibit service name resolution.

AI_V4MAPPED If no IPv6 addresses are found, query for IPv4 addresses and return them to the caller as IPv4-mapped IPv6 ad? dresses.

AI_ALL Query for both IPv4 and IPv6 addresses.

AI_ADDRCONFIG Query for IPv4 addresses only when an IPv4 address is configured; query for IPv6 addresses only when an IPv6 address is configured.

The <netdb.h> header shall define the following symbolic constants that evaluate to bitwise-distinct integer constants for use in the flags ar? gument to getnameinfo():

NI_NOFQDN Only the nodename portion of the FQDN is returned for lo? cal hosts.

NI_NUMERICHOST

The numeric form of the node's address is returned in?

stead of its name.

NI_NAMEREQD Return an error if the node's name cannot be located in the database.

NI_NUMERICSERV

The numeric form of the service address is returned in? stead of its name.

NI_NUMERICSCOPE

For IPv6 addresses, the numeric form of the scope identi? fier is returned instead of its name.

NI_DGRAM Indicates that the service is a datagram service (SOCK_DGRAM).

Address Information Errors

The <netdb.h> header shall define the following symbolic constants for use as error values for getaddrinfo() and getnameinfo(). The values shall be suitable for use in #if preprocessing directives.

EAI_AGAIN The name could not be resolved at this time. Future at? tempts may succeed.

EAI_BADFLAGS The flags had an invalid value.

EAI_FAIL A non-recoverable error occurred.

EAI_FAMILY The address family was not recognized or the address length was invalid for the specified family.

EAI_MEMORY There was a memory allocation failure.

EAI_NONAME The name does not resolve for the supplied parameters. NI_NAMEREQD is set and the host's name cannot be located, or both nodename and servname were null.

EAI_SERVICE The service passed was not recognized for the specified socket type.

EAI_SOCKTYPE The intended socket type was not recognized.

EAI_SYSTEM A system error occurred. The error code can be found in errno.

EAI_OVERFLOW An argument buffer overflowed.

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

```
void             endhostent(void);
void             endnetent(void);
void             endprotoent(void);
void             endservent(void);
void             freeaddrinfo(struct addrinfo *);
const char       *gai_strerror(int);
int              getaddrinfo(const char *restrict, const char *restrict,
                    const struct addrinfo *restrict,
                    struct addrinfo **restrict);
struct hostent   *gethostent(void);
int              getnameinfo(const struct sockaddr *restrict, socklen_t,
                    char *restrict, socklen_t, char *restrict,
                    socklen_t, int);
struct netent    *getnetbyaddr(uint32_t, int);
struct netent    *getnetbyname(const char *);
struct netent    *getnetent(void);
struct protoent  *getprotobyname(const char *);
struct protoent  *getprotobynumber(int);
struct protoent  *getprotoent(void);
struct servent   *getservbyname(const char *, const char *);
struct servent   *getservbyport(int, const char *);
struct servent   *getservent(void);
void             sethostent(int);
void             setnetent(int);
void             setprotoent(int);
void             setservent(int);
```

The <netdb.h> header shall define the socklen_t type through typedef, as described in <sys/socket.h>.

Inclusion of the <netdb.h> header may also make visible all symbols from <netinet/in.h>, <sys/socket.h>, and <inttypes.h>.

The following sections are informative.

## APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

<inttypes.h>, <netinet_in.h>, <sys_socket.h>

The System Interfaces volume of POSIX.1?2017, bind(), endhostent(), endnetent(), endprotoent(), endservent(), freeaddrinfo(), gai_str? error(), getnameinfo()

COPYRIGHT