The Open Group Base Specifications Issue 7, 2018 edition
IEEE Std 1003.1-2017 (Revision of IEEE Std 1003.1-2008)
Copyright © 2001-2018 IEEE and The Open Group

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 include 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.

Address Information Structure

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 addresses.

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* argument to *getnameinfo*():

NI_NOFQDN
: Only the nodename portion of the FQDN is returned for local hosts.

NI_NUMERICHOST
: The numeric form of the node's address is returned instead 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 instead of its name.

NI_NUMERICSCOPE
: For IPv6 addresses, the numeric form of the scope identifier 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 attempts 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>*

XSH *bind*, *endhostent*, *endnetent*, *endprotoent*, *endservent*, *freeaddrinfo*, *gai_strerror*, *getnameinfo*

CHANGE HISTORY

First released in Issue 6. Derived from the XNS, Issue 5.2 specification.

The Open Group Base Resolution bwg2001-009 is applied, which changes the return type for *gai_strerror()* from **char *** to **const char ***. This is for coordination with the IPnG Working Group.

IEEE Std 1003.1-2001/Cor 1-2002, item XBD/TC1/D6/11 is applied, adding a description of the NI_NUMERICSCOPE macro and correcting the *getnameinfo()* function prototype. These changes are for alignment with IPv6.

Issue 7

SD5-XBD-ERN-14 is applied, changing the description of the *s_port* member of the **servent** structure.

The obsolescent *h_errno* external integer, and the obsolescent *gethostbyaddr*() and *gethostbyname*() functions are removed, along with the HOST_NOT_FOUND, NO_DATA, NO_RECOVERY, and TRY_AGAIN macros.

This reference page is clarified with respect to macros and symbolic constants.

End of informative text.

return to top of page

UNIX ® is a registered Trademark of The Open Group.
POSIX ™ is a Trademark of The IEEE.
Copyright © 2001-2018 IEEE and The Open Group, All Rights Reserved
[Main Index | XBD | XSH | XCU | XRAT]

<<< Previous Home Next >>>