The **hostent** structure is used by functions to store information about a given host, such as host name, IPv4 address, and so forth. An application should never attempt to modify this structure or to free any of its components. Furthermore, only one copy of the **hostent** structure is allocated per thread, and an application should therefore copy any information that it needs before issuing any other Windows Sockets API calls.

**Syntax**

C++

typedef struct hostent {

char FAR      \*h\_name;

char FAR FAR \*\*h\_aliases;

short         h\_addrtype;

short         h\_length;

char FAR FAR \*\*h\_addr\_list;

} HOSTENT, \*PHOSTENT, FAR \*LPHOSTENT;

**Members**

**h\_name**

The official name of the host (PC). If using the DNS or similar resolution system, it is the Fully Qualified Domain Name (FQDN) that caused the server to return a reply. If using a local hosts file, it is the first entry after the IPv4 address.

**h\_aliases**

A **NULL**-terminated array of alternate names.

**h\_addrtype**

The type of address being returned.

**h\_length**

The length, in bytes, of each address.

**h\_addr\_list**

A **NULL**-terminated list of addresses for the host. Addresses are returned in network byte order. The macro **h\_addr** is defined to be h\_addr\_list[0] for compatibility with older software.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The **in\_addr** structure represents an IPv4 Internet address.

**Syntax**

C++

typedef struct in\_addr {

union {

struct {

u\_char s\_b1,s\_b2,s\_b3,s\_b4;

} S\_un\_b;

struct {

u\_short s\_w1,s\_w2;

} S\_un\_w;

u\_long S\_addr;

} S\_un;

} IN\_ADDR, \*PIN\_ADDR, FAR \*LPIN\_ADDR;

**Members**

**S\_un**

**S\_un\_b**

An IPv4 address formatted as four **u\_char**s.

**S\_un\_w**

An IPv4 address formatted as two **u\_short**s.

**S\_addr**

An IPv4 address formatted as a **u\_long**.

**Remarks**

The **in\_addr** structure is used with IPv4 addresses.

The **in\_addr** structure is the IPv4 equivalent of the IPv6-based [**in6\_addr**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms738560%28v=vs.85%29.aspx) structure.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**gethostbyname** function retrieves host information corresponding to a host name from a host database.

struct hostent\* FAR gethostbyname(

\_\_in  const char \*name

);

**Parameters**

*name* [in]

A pointer to the **null**-terminated name of the host to resolve.

The **gethostbyname** function returns a pointer to a [**hostent**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms738552%28v=vs.85%29.aspx) structure—a structure allocated by Windows Sockets. The **hostent** structure contains the results of a successful search for the host specified in the name parameter.

If the host specified in the name parameter has both IPv4 and IPv6 addresses, only the IPv4 addresses will be returned. The **gethostbyname** function can only return IPv4 addresses for the name parameter.

The **gethostbyaddr** function retrieves the host information corresponding to a network address.

**Syntax**

C++

struct hostent\* FAR gethostbyaddr(

\_\_in  const char \*addr,

\_\_in  int len,

\_\_in  int type

);

**Parameters**

*addr* [in]

A pointer to an address in network byte order.

*len* [in]

The length, in bytes, of the address.

*type* [in]

The type of the address, such as the AF\_INET

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

inet\_aton()

convert Internet dot address to network address

#include <arpa/inet.h>

int inet\_aton(const char \*cp, struct in\_addr \*addr);

The inet\_ntoa function converts an (Ipv4) Internet network address into an ASCII string in Internet standard dotted-decimal format.

char\* FAR inet\_ntoa(

\_\_in struct in\_addr in

);

Parameters

in [in]