The **connect** function establishes a connection to a specified socket.

**Syntax**

C++

int connect(

\_\_in  SOCKET s,

\_\_in  const struct sockaddr \*name,

\_\_in  int namelen

);

**Parameters**

*s* [in]

A descriptor identifying an unconnected socket.

*name* [in]

A pointer to the [**sockaddr**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740496%28v=vs.85%29.aspx) structure to which the connection should be established.

*namelen* [in]

The length, in bytes, of the [**sockaddr**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740496%28v=vs.85%29.aspx) structure pointed to by the *name* parameter.

The **socket** function creates a socket that is bound to a specific transport service provider.

**Syntax**

C++

SOCKET WSAAPI socket(

\_\_in  int af,

\_\_in  int type,

\_\_in  int protocol

);

**Parameters**

*af* [in]

The address family specification. Possible values for the address family are defined in the *Winsock2.h* header file.

**InetPton** function converts an IPv4 or IPv6 Internet network address in its standard text presentation form into its numeric binary form. The ANSI version of this function is **inet\_pton**.

**Syntax**

C++

INT WSAAPI InetPton(

\_\_in   INT Family,

\_\_in   PCTSTR pszAddrString,

\_\_out  PVOID pAddrBuf

);

**Parameters**

*Family* [in]

The address family.

Possible values for the address family are defined in the *Ws2def.h* header file. Note that the *Ws2def.h* header file is automatically included in *Winsock2.h*, and should never be used directly. Note that the values for the AF\_ address family and PF\_ protocol family constants are identical (for example, **AF\_INET** and **PF\_INET**), so either constant can be used.

The values currently supported are **AF\_INET** and **AF\_INET6**.

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| **AF\_INET**  2 | The Internet Protocol version 4 (IPv4) address family. When this parameter is specified, the *pszAddrString* parameter must point to a text representation of an IPv4 address and the *pAddrBuf* parameter returns a pointer to an [**IN\_ADDR**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms738571%28v=vs.85%29.aspx) structure that represents the IPv4 address. |
| **AF\_INET6**  23 | The Internet Protocol version 6 (IPv6) address family. When this parameter is specified, the *pszAddrString* parameter must point to a text representation of an IPv6 address and the *pAddrBuf* parameter returns a pointer to an [**IN6\_ADDR**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms738560%28v=vs.85%29.aspx) structure that represents the IPv6 address. |

*pszAddrString* [in]

A pointer to the **NULL**-terminated string that contains the text representation of the IP address to convert to numeric binary form.

When the *Family* parameter is **AF\_INET**, then the *pszAddrString* parameter must point to a text representation of an IPv4 address in standard dotted-decimal notation.

When the *Family* parameter is **AF\_INET6**, then the *pszAddrString* parameter must point to a text representation of an IPv6 address in standard notation.

*pAddrBuf* [out]

A pointer to a buffer in which to store the numeric binary representation of the IP address. The IP address is returned in network byte order.

When the *Family* parameter is **AF\_INET**, this buffer should be large enough to hold an [**IN\_ADDR**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms738571%28v=vs.85%29.aspx) structure.

When the *Family* parameter is **AF\_INET6**, this buffer should be large enough to hold an [**IN6\_ADDR**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms738560%28v=vs.85%29.aspx) structure.

The **listen** function places a socket in a state in which it is listening for an incoming connection.

**Syntax**

C++

int listen(

\_\_in  SOCKET s,

\_\_in  int backlog

);

**Parameters**

*s* [in]

A descriptor identifying a bound, unconnected socket.

*backlog* [in]

The maximum length of the queue of pending connections. If set to **SOMAXCONN**, the underlying service provider responsible for socket *s* will set the backlog to a maximum reasonable value. There is no standard provision to obtain the actual backlog value.

The **bind** function associates a local address with a socket.

**Syntax**

C++

int bind(

\_\_in  SOCKET s,

\_\_in  const struct sockaddr \*name,

\_\_in  int namelen

);

**Parameters**

*s* [in]

A descriptor identifying an unbound socket.

*name* [in]

A pointer to a [**sockaddr**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740496%28v=vs.85%29.aspx) structure of the local address to assign to the bound socket .

*namelen* [in]

The length, in bytes, of the value pointed to by the *name* parameter.

The **accept** function permits an incoming connection attempt on a socket.

**Syntax**

C++

SOCKET accept(

\_\_in     SOCKET s,

\_\_out    struct sockaddr \*addr,

\_\_inout  int \*addrlen

);

**Parameters**

*s* [in]

A descriptor that identifies a socket that has been placed in a listening state with the [**listen**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms739168%28v=vs.85%29.aspx) function. The connection is actually made with the socket that is returned by **accept**.

*addr* [out]

An optional pointer to a buffer that receives the address of the connecting entity, as known to the communications layer. The exact format of the *addr* parameter is determined by the address family that was established when the socket from the [**sockaddr**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740496%28v=vs.85%29.aspx) structure was created.

*addrlen* [in, out]

An optional pointer to an integer that contains the length of structure pointed to by the *addr* parameter.

# sockaddr\_in

In the Internet address family, this structure is used by Windows Sockets to specify a local or remote endpoint address to which to connect a socket. This is the form of the [sockaddr](http://msdn.microsoft.com/en-us/library/aa921121.aspx) structure specific to the Internet address family and can be cast to **sockaddr**.

[Syntax](javascript:void(0))

struct sockaddr\_in{

short sin\_family;

unsigned short sin\_port;

IN\_ADDR sin\_addr;

char sin\_zero[8];

};

[Members](javascript:void(0))

**sin\_family**

Address family; must be AF\_INET.

**sin\_port**

Internet Protocol (IP) port.

**sin\_addr**

IP address in network byte order.

**sin\_zero**

Padding to make structure the same size as **SOCKADDR**.

[Remarks](javascript:void(0))

The IP address part of this structure is of type [in\_addr](http://msdn.microsoft.com/en-us/library/aa922011.aspx).

[Requirements](javascript:void(0))

|  |  |
| --- | --- |
| Header | winsock2.h |
| Windows Embedded CE | Windows CE 1.0 and later |
| Windows Mobile | Windows Mobile Version 5.0 and later |

# memset

<cstring>

void \* memset ( void \* ptr, int value, size\_t num );

Fill block of memory

Sets the first *num* bytes of the block of memory pointed by *ptr* to the specified *value* (interpreted as an unsigned char).

### Parameters

ptr

Pointer to the block of memory to fill.

value

Value to be set. The value is passed as an int, but the function fills the block of memory using the *unsigned char* conversion of this *value*.

num

Number of bytes to be set to the value.

### Return Value

*ptr* is returned.