Chapter 21 Winsock Error Codes

This chapter lists Winsock error codes by error number. The list does not include the Winsock errors marked BSD-specific or undocumented. In addition, the Winsock errors that map directly to Windows errors appear toward the end of the chapter.

10004—WSAEINTR

Interrupted function call. This error indicates that a blocking call was interrupted by a call to WSACancelBlockingCall.

10009—WSAEBADF

Bad file handle. This error means that the supplied file handle is invalid. Under Microsoft Windows CE, it is possible for the socket function to return this error, which indicates that the shared serial port is busy.

10013—WSAEACCES

Permission denied. An attempt was made to manipulate the socket, which is forbidden. This error most commonly occurs when attempting to use a broadcast address in sendto or WSASendTo, in which broadcast permission has not been set with setsockopt and the SO_BROADCAST option.

10014—WSAEFAULT

Invalid address. The pointer address passed into the Winsock function is invalid. This error is also generated when the specified buffer is too small.

10022—WSAEINVAL

Invalid argument. An invalid argument was specified. For example, specifying an invalid control code to WSAIoctl generates this error. This code can also indicate an error with the current state of a socket—for example, calling accept or WSAAccept on a socket that is not listening.

10024—WSAEMFILE

Too many open files. Too many sockets are open. Typically, Microsoft providers are limited only by the amount of resources available on the system.

10035—WSAEWOULDBLOCK

Resource temporarily unavailable. This error is most commonly returned on nonblocking sockets in which the requested operation cannot complete immediately. For example, calling connect on a nonblocking socket returns this error because the connection request cannot be completed immediately.

10036—WSAEINPROGRESS

Operation now in progress. A blocking operation is currently executing. Typically, you do not see this error unless you are developing 16-bit Winsock applications.

10037—WSAEALREADY

Operation already in progress. This error typically occurs when an operation that is already in progress is attempted on a nonblocking socket—for example, calling connect or WSAConnect a second time on a nonblocking socket already in the process of connecting. This error can also occur when a service provider is in the process of executing a callback function (for those Winsock functions that support callback routines).

10038—WSAENOTSOCK

Socket operation on an invalid socket. This error can be returned from any Winsock function that takes a SOCKET handle as a parameter. This error indicates that the supplied socket handle is not valid.

10039—WSAEDESTADDRREQ

Destination address required. This error indicates that the supplied address was omitted. For instance, calling sendto with the destination address INADDR_ANY returns this error.

10040—WSAEMSGSIZE

Message too long. This error can mean a number of things. If a message is sent on a datagram socket that is too large for the internal buffer, this error occurs. It also occurs if the message is too large because of a network limitation. Finally, if on receiving a datagram the buffer is too small to receive the message, this error is generated.

10041—WSAEPROTOTYPE

Wrong protocol type for socket. A protocol was specified in a call to socket or WSASocket that does not support the semantics of the given socket type. For example, requesting the creation of an IP socket of type SOCK_STREAM and protocol IPPROTO_UDP generates this error.

10042—WSAENOPROTOOPT

Bad protocol option. An unknown, unsupported, or invalid socket option or level was specified in a call to getsockopt or setsockopt.

10043—WSAEPROTONOSUPPORT

Protocol not supported. Either the requested protocol is not installed on the system or no implementation exists for it. For example, if TCP/IP is not installed on the system, attempting to create either a TCP or UDP socket generates this error.

10044—WSAESOCKTNOSUPPORT

Socket type not supported. Support for the specified socket type does not exist for the given address family. For example, requesting a socket of type SOCK_RAW for a protocol that does not support raw sockets generates this error.

10045—WSAEOPNOTSUPP

Operation not supported. The attempted operation is not supported for the referenced object. Typically, this occurs when trying to call a Winsock function on a socket that does not support that operation. For example, calling accept or WSAAccept on a datagram socket causes this error.

10046—WSAEPFNOSUPPORT

Protocol family not supported. The requested protocol family does not exist or is not installed on the system. In most cases, this error is interchangeable with WSAEAFNOSUPPORT, which occurs more often.

10047—WSAEAFNOSUPPORT

Address family does not support requested operation. This error occurs when attempting to perform an operation that is not supported by the socket type. For example, trying to call sendto or WSASendTo with a socket of type SOCK_STREAM generates this error. This error can also occur when calling socket or WSASocket and requesting an invalid combination of address family, socket type, and protocol.

10048—WSAEADDRINUSE

Address already in use. Under normal circumstances, only one socket is permitted to use each socket address. (For example, an IP socket address consists of the local IP address and port number.) This error is usually associated with the bind, connect, and WSAConnect functions. The socket option SO_REUSEADDR can be set with the setsockopt function to allow multiple sockets access to the same local IP address and port. (For more information, see Chapter 7.)

10049—WSAEADDRNOTAVAIL

Cannot assign requested address. This error occurs when the address specified in an API call is not valid for that function. For example, specifying an IP address in bind that does not correspond to a local IP interface generates this error. This error can also occur when specifying port 0 for the remote machine to connect to with connect, WSAConnect, sendto, WSASendTo, and WSAJoinLeaf.

10050—WSAENETDOWN

Network is down. The operation encountered a dead network. This could indicate the failure of the network stack, the network interface, or the local network.

10051—WSAENETUNREACH

Network is unreachable. An operation was attempted to an unreachable network. This indicates that the local host does not know how to reach the remote host—in other words, no known route to the destination exists.

10052—WSAENETRESET

Network dropped the connection on reset. The connection has been broken because keepalives have detected a failure. This error can also occur when attempting to set the SO_KEEPALIVE option with setsockopt on a connection that has already failed.

10053—WSAECONNABORTED

Software caused the connection to abort. An established connection was aborted due to a software error. Typically, this means the connection was aborted due to a protocol or timeout error.

10054—WSAECONNRESET

Connection reset by peer. The remote host forcibly closed an established connection. This error can occur if the remote process is abnormally terminated (as in memory violation or hardware failure) or if a hard close was performed on the socket. A socket can be configured for a hard close using the SO_LINGER socket option and setsockopt. (For more information, see Chapter 7.)

10055—WSAENOBUFS

No buffer space available. The requested operation could not be performed because the system lacked sufficient buffer space.

10056—WSAEISCONN

Socket is already connected. A connection is being attempted on a socket that is already connected. This can occur on both datagram and stream sockets. When using datagram sockets, if connect or WSAConnect has been called to associate an endpoint's address for datagram communication, attempting to call either sendto or WSASendTo generates this error.

10057—WSAENOTCONN

Socket is not connected. This error occurs when a request is made to send or receive data on a connection-oriented socket that is not currently connected.

10058—WSAESHUTDOWN

Cannot send after socket shutdown. The socket has already been partially closed by a call to shutdown, and either a send or a receive operation is being requested. Note that this occurs only on the data-flow direction that has been shut down. For example, after calling shutdown on sends, any call to send data generates this error.

10060—WSAETIMEDOUT

Connection timed out. This error occurs when a connection request has been made and the remote computer fails to properly respond (or doesn't respond at all) after a specified length of time. This error is typically seen when the socket options SO_SNDTIMEO and SO_RCVTIMEO are set on a socket as well as when the connect and WSAConnect functions are called. For more information on setting SO_SNDTIMEO and SO_RCVTIMEO on a socket, see Chapter 7.

10061—WSAECONNREFUSED

Connection refused. The connection could not be established because the target machine refused it. This error usually occurs because no application on the remote machine is servicing connections on that address.

10064—WSAEHOSTDOWN

Host is down. This error indicates that the operation has failed because the destination host is down; however, an application is more likely to receive the error WSAETIMEDOUT because it typically occurs when attempting to establish a connection.

10065—WSAEHOSTUNREACH

No route to host. An operation was attempted to an unreachable host. This error is similar to WSAENETUNREACH.

10067—WSAEPROCLIM

Too many processes. Some Winsock service providers set a limit on the number of processes that can simultaneously access them.

10091—WSASYSNOTREADY

Network subsystem is unavailable. This error is returned when calling WSAStartup, and the provider cannot function because the underlying system that provides services is unavailable.

10092—WSAVERNOTSUPPORTED

Winsock.dll version out of range. The requested version of the Winsock provider is not supported.

10093—WSANOTINITIALISED

Winsock has not been initialized. A successful call to WSAStartup has not yet been performed.

10101—WSAEDISCON

Graceful shutdown in progress. WSARecv and WSARecvFrom return this error to indicate that the remote party has initiated a graceful shutdown. This error occurs on message-oriented protocols such as ATM.

10102—WSAENOMORE

No more records found. WSALookupServiceNext returns this record to indicate that no additional records are left. This error is interchangeable with WSA_E_NO_MORE. Applications should check for both this error and WSA_E_NO_MORE.

10103—WSAECANCELLED

Operation canceled. This error indicates that a call to WSALookupServiceEnd was made while a call to WSALookupServiceNext was still processing. WSALookupServiceNext returns this error. This code is interchangeable with WSA_E_CANCELLED. Applications should check for both this error and WSA_E_CANCELLED.

10104—WSAEINVALIDPROCTABLE

The procedure call table is invalid. A service provider typically returns this error when the procedure table contains invalid entries. For more information on service providers, see Chapter 12.

10105—WSAEINVALIDPROVIDER

Invalid service provider. This error is associated with service providers, and it occurs when the provider cannot establish the correct Winsock version needed to function correctly.

10106—WSAEPROVIDERFAILEDINIT

The provider failed to initialize. This error is associated with service providers, and it is typically seen when the provider cannot load the necessary DLLs.

10107—WSASYSCALLFAILURE

System call failure. A system call that should never fail has failed.

10108—WSASERVICE_NOT_FOUND

No such service found. This error is normally associated with registration and name resolution functions when querying for services. (See Chapter 8 for more information about these functions.) This error indicates that the requested service could not be found in the given namespace.

10109—WSATYPE_NOT_FOUND

Class type not found. This error is also associated with the registration and name resolution functions when manipulating service classes. When an instance of a service is registered, it must reference a service class that was previously installed with WSAInstallServiceClass.

10110—WSA_E_NO_MORE

No more records found. This error is returned from WSALookupServiceNext to indicate that no additional records are left. It is interchangeable with WSAENOMORE. Applications should check for both this error and WSAENOMORE.

10111—WSA_E_CANCELLED

Operation canceled. This error indicates that a call to WSALookupServiceEnd was made while a call to WSALookupServiceNext was still processing. WSALookupServiceNext returns this error. This code is interchangeable with

WSAECANCELLED. Applications should check for both this error and WSAECANCELLED.

10112—WSAEREFUSED

Query refused. A database query failed because it was actively refused.

11001—WSAHOST_NOT_FOUND

Host not found. This error occurs with gethostbyname and gethostbyaddr to indicate that an authoritative answer host was not found.

11002—WSATRY_AGAIN

Nonauthoritative host not found. This error is also associated with gethostbyname and gethostbyaddr, and it indicates that either the nonauthoritative host was not found or a server failure occurred.

11003—WSANO_RECOVERY

A nonrecoverable error occurred. This error is also associated with gethostbyname and gethostbyaddr. It indicates that a nonrecoverable error has occurred and the operation should be tried again.

11004—WSANO_DATA

No data record of the requested type found. This error is also associated with gethostbyname and gethostbyaddr. It indicates that the supplied name was valid but no data record of the requested type was found with it.

11005—WSA_QOS_RECEIVERS

At least one reserve message has arrived. This value is associated with IP Quality of Service (QOS) and is not an error per se. (See Chapter 10 for more on QOS.) It indicates that at least one process on the network is interested in receiving QOS traffic.

11006—WSA_QOS_SENDERS

At least one path message has arrived. This value is associated with QOS and is more of a status message. This value indicates that at least one process on the network is interested in sending QOS traffic.

11007—WSA_QOS_NO_SENDERS

No QOS senders. This value is associated with QOS and indicates that there are no longer any processes interested in sending QOS data. See Chapter 10 for a more complete description of when this error occurs.

11008—WSA_QOS_NO_RECEIVERS

No QOS receivers. This value is associated with QOS and indicates that there are no longer any processes interested in receiving QOS data. See Chapter 10 for a more complete description of this error.

11009—WSA_QOS_REQUEST_CONFIRMED

Reservation request has been confirmed. QOS applications can request that they be notified when their reservation request for network bandwidth has been approved. When such a request is made, this is the message generated. See Chapter 10 for a more complete description.

11010—WSA_QOS_ADMISSION_FAILURE

Error due to lack of resources. Insufficient resources were available to satisfy the QOS bandwidth request.

11011—WSA_QOS_POLICY_FAILURE

Invalid credentials. Either the user did not possess the correct privileges or the supplied credentials were invalid when making a QOS reservation request.

11012—WSA_QOS_BAD_STYLE

Unknown or conflicting style. QOS applications can establish different filter styles for a given session. This error indicates either unknown or conflicting style types. See Chapter 10 for a description of filter styles.

11013—WSA_QOS_BAD_OBJECT

Invalid FILTERSPEC structure or provider-specific object. This error occurs if either the FLOWSPEC structures or the provider-specific buffers of a QOS object are invalid. See Chapter 10 for more details.

11014—WSA_QOS_TRAFFIC_CTRL_ERROR

Problem with a FLOWSPEC. This error occurs if the traffic control component has a problem with the supplied FLOWSPEC parameters that are passed as a member of a QOS object.

11015—WSA_QOS_GENERIC_ERROR

General QOS error. This is a catch-all error that is returned when the other QOS errors do not apply.

6—WSA_INVALID_HANDLE

Specified event object invalid. This Windows error is seen when using Winsock functions that map to Win32 functions. This particular error occurs when a handle passed to WSAWaitForMultipleEvents is invalid.

8—WSA_NOT_ENOUGH_MEMORY

Insufficient memory available. This Windows error indicates that insufficient memory is available to complete the operation.

87—WSA_INVALID_PARAMETER

One or more parameters are invalid. This Windows error indicates that a parameter passed into the function is invalid. This error also occurs with WSAWaitForMultipleEvents when the event count parameter is not valid.

258—WSA_WAIT_TIMEOUT

Operation timed out. This Windows error indicates that the overlapped operation did not complete in the specified time.

995—WSA_OPERATION_ABORTED

Overlapped operation aborted. This Windows error indicates that an overlapped I/O operation was canceled because of the closure of a socket. In addition, this error can occur when executing the SIO_FLUSH ioctl command.

996—WSA_IO_INCOMPLETE

Overlapped I/O event object is not in a signaled state. This Windows error is also associated with overlapped I/O. It is seen when calling WSAGetOverlappedResults and indicates that the overlapped I/O operation has not yet completed.

997—WSA_IO_PENDING

Overlapped operations will complete later. When making an overlapped I/O call with a Winsock function, this Windows error is returned to indicate that the operation is pending and will complete later. See Chapter 5 for a discussion of overlapped I/O.