

Socket Error.txt

Socket error 0 - Directly send error
Socket error 10004 - Interrupted function call
Socket error 10013 - Permission denied
Socket error 10014 - Bad address
Socket error 10022 - Invalid argument
Socket error 10024 - Too many open files
Socket error 10035 - Resource temporarily unavailable
Socket error 10036 - Operation now in progress
Socket error 10037 - Operation already in progress
Socket error 10038 - Socket operation on non-socket
Socket error 10039 - Destination address required
Socket error 10040 - Message too long
Socket error 10041 - Protocol wrong type for socket
Socket error 10042 - Bad protocol option
Socket error 10043 - Protocol not supported
Socket error 10044 - Socket type not supported
Socket error 10045 - Operation not supported
Socket error 10046 - Protocol family not supported
Socket error 10047 - Address family not supported by protocol family
Socket error 10048 - Address already in use
Socket error 10049 - Cannot assign requested address
Socket error 10050 - Network is down
Socket error 10051 - Network is unreachable
Socket error 10052 - Network dropped connection on reset
Socket error 10053 - Software caused connection abort
Socket error 10054 - Connection reset by peer
Socket error 10055 - No buffer space available
Socket error 10056 - Socket is already connected
Socket error 10057 - Socket is not connected
Socket error 10058 - Cannot send after socket shutdown
Socket error 10060 - Connection timed out
Socket error 10061 - Connection refused
Socket error 10064 - Host is down
Socket error 10065 - No route to host
Socket error 10067 - Too many processes
Socket error 10091 - Network subsystem is unavailable
Socket error 10092 - WINSOCK.DLL version out of range
Socket error 10093 - Successful WSStartup not yet performed
Socket error 10094 - Graceful shutdown in progress
Socket error 11001 - Host not found
Socket error 11002 - Non-authoritative host not found
Socket error 11003 - This is a non-recoverable error
Socket error 11004 - Valid name, no data record of requested type

WSAEADDRINUSE (10048) Address already in use
WSAECONNABORTED (10053) Software caused connection abort
WSAECONNREFUSED (10061) Connection refused
WSAECONNRESET (10054) Connection reset by peer
WSAEDESTADDRREQ (10039) Destination address required
WSAEHOSTUNREACH (10065) No route to host
WSAEMFILE (10024) Too many open files
WSAENETDOWN (10050) Network is down
WSAENETRESET (10052) Network dropped connection
WSAENOBUFS (10055) No buffer space available
WSAENETUNREACH (10051) Network is unreachable
WSAETIMEDOUT (10060) Connection timed out

Socket Error.txt

WSAHOST_NOT_FOUND (11001) Host not found
WSASYSNOTREADY (10091) Network sub-system is unavailable
WSANOTINITIALISED (10093) WSASStartup() not performed
WSANO_DATA (11004) Valid name, no data of that type
WSANO_RECOVERY (11003) Non-recoverable query error
WSATRY_AGAIN (11002) Non-authoritative host found
WSAVERNOTSUPPORTED (10092) Wrong WinSock DLL version
+++++

常見SOCKET錯誤返回碼（轉）

WSAEINTR (10004) 被中斷的系統呼叫

當以阻攔式進行的WinSock函式被WSACancelBlockingCall()中斷的時候，這個阻攔式函式會得到WSAEINTR這個錯誤訊息。讀者要注意的是，當你的程式有用WSACancelBlockingCall去中斷阻攔式函式的時候，你必須為這個阻攔式函式寫處理WSAEINTR錯誤訊息的程式碼，否則你的程式可能會出現些無預期的錯誤。

原則上，所有能以阻攔式進行的函式都可能會發生這個錯誤。

WSAEBADF (10009) 錯誤的檔案代碼

柏克萊socket介面中，檔案描述子和socket描述子是相通的，開啟socket可以想像成開啟一個檔案。WSAEBADF在柏克萊socket介面的意義是指錯誤的socket描述子，這個錯誤的socket描述子可能是未經開啟的socket或是以關閉的socket。

在WinSock中有個相通的錯誤代碼WSAENOTSOCK，用來指定錯誤的socket描述子。詳細說明請參考WSAENOTSOCK部分。

WSAEACCES (10013) 無此權限

對於柏克萊socket介面函式，這個錯誤發生的原因通常是開啟一個不具有開啟權限的檔案或socket。例如在UNIX上，一般的使用者不能開啟SOCK_RAW的socket，通常是超級使用者(super user)才有此權限。如果一般的使用者企圖開啟SOCK_RAW的socket，將會得到這個錯誤訊息。

對於WinSock API函式介面，發生此錯誤的函式有二：send()和sendto()。當利用send()或sendto()傳送資料的時候，將目的位址設成廣播位址(broadcast address)，但是並未呼叫setsockopt()設定SO_BROADCAST選項時便會發生WSAEACCES錯誤。

WSAEFAULT (10014) 錯誤的記憶體位址

系統偵測出使用錯誤的記憶體位址。呼叫WinSock函式時，常常會有參數是以指標的方式傳入，這個指標可能是指向一個結構，如sockaddr_in結構，或是指向一個資料緩衝區。當程式不小心傳入一個錯誤記憶體未指的指標時，就會發生WSAEFAULT這個錯誤。此外，當指標所指向的記憶體區塊太小的時候，也會發生這個錯誤。

WSAEINVAL (10022) 參數錯誤

如果傳入不正確的參數給WinSock函式的話，會導致WSAEINVAL這個錯誤。如果參數是一個結構的指標，此結構內容填入不正確的值，也會導致這個錯誤。例如，shutdown()函式的how()參數只能是0、1或2，若是設定成其他數值，就會出現這個錯誤。

在網路程式的設計中，函式和函式之間的呼叫通常有一定的順序，如過不照這個順序進行，也會出現這個錯誤。底下是針對每個相關函式的詳細說明。

函式

錯誤發生的時機

accept()

呼叫accept()之前沒有成功呼叫listen()

bind()

已經成功呼叫bind()函式而指定socket的名稱了

getstockname()

沒有呼叫bind()函式指定socket名稱

listen()

已經處於連接狀態或是socket沒有呼叫bind()函式指定socket名稱

recv()和recvfrom()

對於datagram socket, socket沒有呼叫bind()函式指定IP位址、port和協定等；對於stream socket, 連接尚未建立

send()和sendto()

對於datagram socket, socket沒有呼叫bind()函式指定IP位址、port和協定等；對於stream socket, 連接尚未建立

原則上，所有函式其參數所能設定的值有一定的範圍，或是有一些特定的值，這些函式都有可能發生這個錯誤。

WSAEMFILE (10024) 太多開啟的檔案

柏克萊socket介面的解釋是開啟過多的檔案，超過檔案開啟數的限制。在柏克萊socket將檔案和socket的開啟以檔案描述子(file descriptor)描述。WinSock對於socket是有別於檔案的，WSAEMFILE在WinSock的意義是開啟太多的socket。

這個程式通常發生在同一部機器中執行過多的網路程式，以致開啟過多的socket。另一種可以避免的錯誤是應用程式沒有正常的關閉不用的socket，以致socket資源耗盡。

WSAEWOULDBLOCK (10035) 函式作用阻攔中

當函式作用是以非阻攔式進行，而此刻的函式作用再返回時其函式作用尚未完成的時候，就會出現這個錯誤。舉例來說，假設send()以非阻攔式傳送一段資料至遠端主機，假設函式返回時得到WSAEWOULDBLOCK這個錯誤，這樣表示這一段資料不能在這一次的send()函式呼叫中傳送完。

對於connect()函式，這個錯誤表示尚未成功地連接上遠端主機，連接要求正在進行中。若你的程式中有利用setsockopt()函式以SO_LINGER設定時限，在呼叫closesocket()時有可能出現這個錯誤。

對於非同步的網路資料函式(WSAAsyncGetXXXByYYYY())而言，得到WSAEWOULDBLOCK錯誤，表示要查詢的網路資料目前無法取得。這通常是個暫時性的錯誤，應用程式可能稍後再呼叫相同的函式就會成功。

WSAEINPROGRESS (10036) 有阻攔函式正在執行中

對於每個process或thread, WinSock只允許”一”個阻攔式函式的執行。如果已有阻攔式函式正在執行，而又呼叫到WinSock函式，函式會傳回WSAEINPROGRESS的錯誤。

這個錯誤也會發生在需要長執行時間的函式，如connect()。有一種情況是這樣：當執行”

非阻攔式”的connect()函式兩次，由於是非阻攔式的 connect()，第一次的呼叫傳回 WSAEWOULDBLOCK錯誤碼。此時要求連接動作尚未完成，第二次的connect()呼叫就會得到 WSAEINPROGRESS的錯誤。

我們的建議是在有可能發生這種錯誤的函式呼叫前加上WSAIsBlocking()判斷是否有阻攔式函式正在執行。如果有的話，我們可能需要將正在進行的阻攔式函式取消掉，或是將目前想要做的事取消掉。

WSAEALREADY (10037) 函式作用已完成

這個錯誤的意思在柏克萊socket介面和WinSock函式的解釋上有很大的差別。柏克萊socket介面的解釋是函式作用已完成；在WinSock中，WSAEALREADY的意思是：“你想要取消的非同步事件已經被取消了！”，也就是當你呼叫WSACancelAsyncRequest()試圖去取消一個非同步事件時，而該事件已被取消了。

WSAENOTSOCK (10038) 非法的socket

通常WinSock函式都要以socket描述子當參數傳入，如果這個socket描述子所代表的socket不是正確開啟的或是根本沒有開啟，函式就會得到WSAENOTSOCK錯誤。原則上，只要以socket描述子當參數傳入的函式都有發生WSAENOSOCK錯誤的可能。

WSAEDESTADDRREQ (10039) 需要對方位址

當你要傳送資料給遠端機器，必須指定對方的位址(和port等等)，才能正確送達。檢查看看你的應用程式是否在呼叫connect()或sendto()時，正確地填入對方的sockaddr結構。

WSAEMSGSIZE (10040) 訊息太長，緩衝區無法容納

這個錯誤發生原因是由於訊息資料過大，緩衝區無法容納，這裡的緩衝區可能是使用者指定的或是系統內部用的緩衝區。會發生這個錯誤的函式有四：

函式
錯誤發生的時機

recv()、recvform()

利用datagram socket接收資料時，如果recv()、recvform()指定的緩衝區長度小於接收進來datagram，這時後過長的datagram會被丟棄並告訴函式WSAEMSGSIZE這個錯誤

send()、sendto()

利用datagram socket傳送資料時，如果送出的資料緩衝區大於WinSock所能傳遞datagram的最大長度，這時send()和sendto()傳回發生WSAEMSGSIZ這個錯誤訊息

WSAEPROTOTYPE (10041) 錯誤的協定型別

錯誤產生的原因是在socket()函式呼叫時指定了WinSock不支援的協定舉例來說，你不可以 stream socket(SOCK_STREAM)設定以UDP協定傳輸(IPPROTO_UDP)。stream socket並不支援UDP通訊協定。如果你真的有心去測試的話，程式中故意如此呼叫 socket():hSock=socket(AF_INET, SOCK_STREAM, IPPROTO_UDP);你會發現，這樣的socket()函式呼叫會得到WSAEPROTONOSUPPORT錯誤。這顯然不對，照理應該得到 WSAEPROTOTYPE這個錯誤才對。WSAEPROTOTYPE這個錯誤表示指定了錯誤的協定，而所指定的協定是該位址家族(協定家族)所支援的，但不適用於socket型別。以上述的例子來看，UDP協定是屬於AF_INET這個位址家族，但是並不能用於stream socket(SOCK_STREAM socket)上。WSAEPROTONOSUPPORT這個錯誤表示所指定的協定是該協定家族所不支援的，意思上和WSAEPROTOTYPE有點出入。

事實上，WSAEPROTOTYPE和WSAEPROTONOSUPPORT之間的差別是微乎其微的。在發展網路程式的時候，這兩種錯誤都是代表著同一種狀況——協定指定錯誤。

總之，你永遠也沒辦法試出WSAEPROTOTYPE這個錯誤。

WSAENOPROTOOPT (10042) 錯誤的協定選項(option)

這個錯誤主要是針對getsockopt()和setsockopt()函式的。這兩個函式可以以參數設定不同的選項而有不同的作用。對於每個選項多多少少會有些限制，如有些選項只能用在stream socket，而有些只能用在datagram socket。如果違反這些限制，就會發生這個錯誤。請看下表：

SO_DONTLINGER

SO_KEEPAIVE

SO_LINGER

SO_OOBNLINE

TCP_NODELAY

datagram socket (SOCK_DGRAM)不支援這些選項

SO_DEBUG

SO_DONTROUTE

SO_RCVBUF

SO_SNDBUF

TCP_NODELAY

有些廠商的WinSock TCP/IP協定核心不支援這些選項

SO_ACCEPTCONN

SO_ERROR

SO_TYPE

這些選項只適用於getsockopt()函式，不能作用於setsockopt()函式

WSAEPROTONOSUPPORT (10043) 不支援的協定

WinSock協定核心不支援所指定的通訊協定。舉例來說當你用socket()函式時，第二個參數指定為SOCK_STREAM，第三個參數指定為 IPPROTO_ICMP，如此socket()呼叫會得到WSAEPROTONOSUPPORT這個錯誤。WinSock 1.1版並不支援ICMP協定，所以呼叫socket()指定ICMP這個協定，就會發生錯誤。事實上，產生WSAEPROTONOSUPPORT錯誤的狀況不只這樣，詳見WSAEPROTOTYPE。

WSAESOCKTNOSUPPORT (10044) 不支援的socket型別

目前使用的WinSock版本不支援函式所指定的socket型別。舉例來說，完全根據WinSock 1.1版的規格書發展的WinSock協定核心，並不支援SOCK_RAW這個socket型別。在呼叫socket()函式時，位址家族選 AF_INET(PF_INET)而socket型別選SOCK_RAW的話，就會發生這個錯誤。

有些廠商發展的WinSock協定核心除了支援WinSock 1.1版的規格書所規範之外，還提供許多不同的功能。我們建議您在發展程式時，如果有必要用到這些額外的功能，請好好考慮一下和其他環境相容性的問題。

WSAEOPNOTSUPP (10045) 作用函式不支援此類socket

函式作用不支援目前socket型別。舉例來說，accept()和listen()函式只能作用在連接導向的socket(stream socket)，也就是stream socket上。對於recv()、recvform()、send()和sendto()函式，MSG_OOB也是只能用於連接導向的socket。

WSAEPFNOSUPPORT (10046) 不支援的協定家族

目前的WinSock協定核心不支援所指定的協定家族。這個錯誤和WSAEAFNOSUPPORT很像，目前這個錯誤被WSAEAFNOSUPPORT所取代了，讀者可以參考WSAEAFNOSUPPORT說明部分。

WSAEAFNOSUPPORT (10047) 不支援的位址家族

使用協定不支援指定的位址家族。對於socket()函式，錯誤發生原因是函式轉物錯誤的參數。對於以sockaddr結構為參數的函數，錯誤發生的原因常是網路協定不支援sockaddr結構中(sin_family成員)指定的位址家族。

WSAEADDRINUSE (10048) 位址已被指定了

所指定的”位址”正在使用中。這裡的位址是指socket的名稱，包含三部分：協定、port和IP位址。這個錯誤發生的原因通常是應用程式試圖利用bind()函式去指定port時，此port已被其他應用程式所佔用了。這種情況發生於在同一部主機上執行兩個相同性質(指定相同socket名稱)的server程式，第二的程式會因為第一個程式先佔用某個port而得到此錯誤訊息。

還有一種常見的情況就是在撰寫client程式時，利用bind()去指定socket的port，此舉常會有這個錯誤發生。並非不能用bind()去指定socket的port，而是你無法確定程式執行的時候，所指定的port是否有其他程式正在使用。通常很少會在client程式中呼叫bind()函式，一般的client程式都是讓系統幫你選一個適當的port。

此外，應用程式可以利用setsockopt()的SO_REUSEADDR選項，允許port可以重複使用，可是我們並不鼓勵這樣做。這樣可能會造成TCP/IP核心在判斷應用程式上的困擾。

WSAEADDRNOTAVAIL (10049) 無法指定位址

錯誤發生的原因通常是指定了該部機器不應該有的位址。這裡的位址包含三個部分：網路協定、port和IP位址。

在柏克萊socket介面中，呼叫bind()時，若指定的IP位址不是本地機器該有的位址，就會出現這個錯誤。在呼叫connect()或sendto()函式時，如果sockaddr_in結構中的sin_port填”0”，如此也會導致這個錯誤(在UNIX上是EADDRNOTAVAIL錯誤)。

對於WinSock函式，錯誤發生的時機和柏克萊socket函式類似，bind()和connect()都會導致這樣的錯誤。

對於WinSock函式，錯誤發生的時機和柏克萊socket函式類似，bind()和connect()都會導致這樣的錯誤。WinSock 1.1版的規格書中並沒有提到bind()函式會導致WSAEADDRNOTAVAIL錯誤，不過根據我們測試的結果，bind()也會導致這樣的錯誤。

WSAENETDOWN (10050) 網路出錯

下層的網路系統發生問題，WinSock函式無法執行。這是一個相當嚴重的問題，這會使得幾乎所有的WinSock函式呼叫失敗。若你的WinSock函式發生錯誤，那麼請你該好好地檢查你的網路卡，網路卡驅動程式和你的WinSock協定核心。

WSAENETUNREACH (10051) 無法連上指定的網路

在柏克萊socket介面中，這個錯誤的意思是無法和對方網路系統取得聯繫。在WinSock 1.1規格中，WSAENETUNREACH相通於柏克萊socket介面中的ETHOSTUNREACH。

WSAENETRESET (10052) 網路已被重新設定

對方主機出錯而重新開機。在本地主機呼叫和傳送資料有關的函式就會得到這個錯誤。對於setsockopt()函式，這個錯誤發生的原因在於試圖設定SO_KEEPALIVE於逾時(timeout)的連接。

WSAECONNABORTED (10053) 中斷連接

在連接建立後，當有某種原因導致TCP/IP協定核心不斷地重送資料，而且都重送失敗的時候，TCP/IP協定將連接中斷。此時，應用程式就會收到WSAECONNABORTED這個錯誤。

WSAECONNRESET (10054) 連結已被對方重新設定

連結已被對方系統強迫中斷了。TCP/IP協定核心如果偵測出stream socket的連接發生問題的話，它會傳一個表示連接重設(reset)的訊息給對方。此時對方程式會得到WSAECONNRESET的錯誤訊息。

WSAENOBUFS (10055) 緩衝區過小

WinSock核心無法配置足夠的緩衝區，無法執行該程式。這個錯誤的發生和整個Windows系統資源有關。如果整個Windows本身可用的資源就很少，所能同時執行的程式也少。也有可能是執行了“不良”的應用程式，沒有正常的使用和歸還系統資源(如記憶體)。或是某些應用程式的不正常關閉，也會佔用系統資源。

WSAEISCONN (10056) socket已經連接

網路函式通常會有一定的呼叫順序，有的函式呼叫成功後便不能再呼叫，否則會出現錯誤。當連接建立時，client程式呼叫()成功，而server程式呼叫accept()成功。此時若client程式再呼叫一次connect()函式的話，會得到WSAEISCONN這個錯誤，表示該socket已經處於連接狀態了。如果client程式呼叫listen()函式的話也會出現這個錯誤，表示不應該在此時呼叫這個函式。對於server程式，如果socket已經處於連接狀態，程式呼叫connect()函式的話，也會得到WSAEISCONN這個錯誤。

上述錯誤是發生在stream socket上，因為stream socket才會建立連接。對於datagram socket，多次成功的connect()函式呼叫是被允許的(對於datagram socket而言，呼叫connect()函式的目的不在於建立連接)。datagram socket發生此錯誤的原因是在sendto()函式上，當呼叫sendto()傳送datagram時，如果對方位址填入INADDR_ANY (“0.0.0.0”)而port填入0(在sockaddr_in結構中)，就會導致WSAEISCONN這個錯誤。

WSAENOTCONN (10057) socket尚未連接

網路函式通常會有一定的呼叫順序，有些函式必須先呼叫成功某函式才能呼叫。WSAENOTCONN這個錯誤發生的原因很明顯，就是在連接建立之前，你試圖從連接對方取得資料或是傳送資料到對方，就會發生WSAENOTCONN這個錯誤。

舉例來說，對於stream socket，如果在建立連結前呼叫send()試圖傳送資料，或者是呼叫recv()試圖取得資料，就會發生這個錯誤，告訴你socket尚未建立連接。你如果呼叫了getpeername()這個函式試圖去取得連接對方的socket，當然也會出現這個錯誤。對於setsockopt()這個函式，如果設定SO_KEEPALIVE，但連接已被中斷的話，也會出現WSAENETCONN這個錯誤。

初學者可能在呼叫connect()函式之後沒有檢查是否函式呼叫錯誤，以致不知道是否連接已正確建立。

WSAESHUTDOWN (10058) socket已經關閉

這個錯誤碼的意思是，無法再傳送資料了，因為socket已經經由shutdown()函式而關閉了。當你利用shutdown()關閉socket之後，WinSock的TCP/IP協定核心就會關閉這個socket的傳送或接收的功能。有一點要注意的是，這個錯誤只會在stream socket出現。

WSAETOOMANYREFS (10059) 參照(reference)太多，資源耗盡

在UNIX作業系統中的解釋是指系統核心資源消耗殆盡。不過在WinSock 1.1版的規格書中並沒有任何一個函式會發生這種錯誤。

WSAETIMEDOUT (10060) 連接逾時(timeout)

當程式試圖利用connect()連接一台遠端主機時，此主機一直沒有回應，經過了一段時間後，TCP/IP協定核心便會傳回connect()函式 WSAETIMEDOUT這個錯誤訊息。發生這個錯誤的原因可能是程式所要連接的遠端主機的系統發生問題，或者根本沒有這一台主機。也有一種可能的原因是本地機器的網路系統出現問題，導致本地主機無法正常與遠端主機聯繫。檢查你的網路系統的設定，看看有什麼不當的地方，如閘道器(gateway)的設定，主機名稱表(hosts)內主機名稱和網路位址的對照有錯誤等等。

WSAECONNREFUSED (10061) 連接受拒

連接要求受到對方機器的回絕，無法和對方建立連接。這個錯誤並不是網路系統出了問題，而是對方機器的server程式拒絕client程式的連接要求，或者是根本沒有這個server程式存在。當TCP/IP核心收到某個連接要求時，會查看系統中是否有某server程式所監聽的port正好是這個連接要求所要連接的port，如果沒有此server程式，TCP/IP核心會回絕一個WSAECONNREFUSED給對方案式。

通常發生這個錯誤的原因有幾個。使用者可能連接到錯誤的遠端主機或是錯誤的port，也有可能是遠端主機和port對了，但是server程式並沒有執行。檢查看看程式中的sockaddr_in結構有沒有填錯？sockaddr_in中的IP位址和port是不是以網路位元組順序排列。

WSAELOOP (10062) 太多層的符號式鏈結(Symbolic link)

在UNIX作業系統中，這個錯誤的意思是指路徑參考過多的符號式鏈結(symbolic link)。

WSAENAMETOOLONG (10063) 檔名過長

在UNIX作業系統，這個錯誤的意思是指欲開啟檔案的路徑名稱(包括檔案名稱)太長。

WSAEHOSTDOWN (10064) 對方主機關閉

在柏克萊socket介面的解釋中，這個錯誤發生的原因是當遠端主機關閉或系統出問題。

WSAEHOSTUNREACH (10065) 無法聯繫對方主機

在柏克萊socket介面中，這個錯誤的意思是無法和對方主機取得聯繫。在WinSock方面，有

類似的錯誤代碼，讀者可以參考WSAENETUNREACH這個錯誤的說明。

WSAEPROCLIM (10067) Too many processes.

A Windows Sockets implementation may have a limit on the number of applications that may use it simultaneously. WSASStartup() may fail with this error if the limit has been reached.

WSASYSNOTREADY (10091) 主機網路系統尚未備妥

本地主機的網路系統尚未能正常運作。若發生這個錯誤，讀者可能要檢查一下有沒有WINSOCK.DLL (或32位元版本的WSOCK32.DLL) 這個檔案?這個檔存放的路徑正確嗎?

WSAVERNOTSUPPORTED (10092) 不支援的WinSock.DLL版本錯誤發生的原因是在呼叫WSASStartup() 函式時所指定的WinSock版本不被現有的WinSock版本所支援。若發生這個錯誤，讀者要確定一下本地Windows系統中的WinSock版本和該版本所能支援的版本是如何等等。通常較新的版本同時能支援新舊版本的WinSock，建議讀者能使用較新的WinSock版本。

WSANOTINITIALISED (10093) 未執行WSASStartup()

應用程式沒有呼叫WSASStartup() 或是呼叫WSASStartup() 失敗。應用程式要成功地呼叫WSASStartup() 才能使用WinSock.DLL，否則會有此錯誤訊息出現。

WSAEDISCON (10101) Graceful shutdown in progress.

Returned by WSARcv(), WSARcvFrom() to indicate the remote party has initiated a graceful shutdown sequence.

WSATYPE_NOT_FOUND (10109) Class type not found

The specified class was not found.

WSAHOST_NOT_FOUND (11001) 已經證實找不到主機

當你使用網路資料函式尋找主機的資料，而該主機不存在時，就會發生這個錯誤。尋找網路主機資料的方法通常是先尋找本地主機的網路名稱表(host檔)，如果找不到再透過DNS的途徑去找，如果再找不到，通常就會回給函式這樣的錯誤。

發生這樣的錯誤時，檢查看看本地主機的主機名稱解譯的設定有沒有問題。比較可能出現問題的是DNS的設定，確定是不是設定錯誤的DNS server。如果DNS server正確，確定一下DNS server有沒有跑起來。你可以利用ping這個程式看看DNS server有沒有跑起來，如果沒有ping這個程式，你可以試著去解釋一個確定存在的主機，看看DNS server有沒有在正常運作中。

WSATRY_AGAIN (11002) 找不到網路資料，但未被證實

這是一個暫時性的錯誤，告訴你目前找不到所要找的網路資料，你可以再試試看。

WSANO_RECOVERY (11003) 無法挽救的錯誤

對於查詢主機名稱而言，這個錯誤發生的原因是由網域名稱系統(Domain Name System, DNS)所出現的錯誤所引起的。這些錯誤包括DNS的FORMERR、REFUSED和NOTIMP等錯誤，詳細請參考RFC1035有關網域名稱系統的文件。

對於查詢服務和協定資料而言，這個錯誤發生的原因是找不到存放網路資料的資料庫。這個

資料庫通常是在本地主機上的檔案，如SERVICES和PROTOCOL這兩個檔。

WSANO_DATA (11004) 名稱合法，但查無此資料

查詢格式是正確的，但是查無此資料存在。這通常是暫時性的錯誤，可能在不同的網路資料伺服器中能找到這份資料。

WSAPROVIDERFAILEDINIT (OS dependent) Unable to initialize a service provider.

Either a service provider's DLL could not be loaded (LoadLibrary() failed) or the provider's WSPStartup/NSPStartup function failed.

WSASYSCALLFAILURE (OS dependent) System call failure..

Returned when a system call that should never fail does. For example, if a call to WaitForMultipleObjects() fails or one of the registry APIs fails trying to manipulate the

protocol/namespace catalogs.

WSA_INVALID_HANDLE (OS dependent) Specified event object handle is invalid.

An application attempts to use an event object, but the specified handle is not valid.

WSA_INVALID_PARAMETER (OS dependent) One or more parameters are invalid.

An application used a WinSock function which directly maps to a Win32 function. The Win32 function is indicating a problem with one or more parameters.

WSA_INVALIDPROCEDURE (OS dependent) Invalid procedure table from service provider.

A service provider returned a bogus proc table to WS2_32.DLL. (Usually caused by one or more of the function pointers being NULL.)

WSA_INVALIDPROVIDER (OS dependent) Invalid service provider version number.

A service provider returned a version number other than 2.2.

WSA_IO_INCOMPLETE (OS dependent) Overlapped I/O event object not in signaled state.

The application has tried to determine the status of an overlapped operation which is not yet completed. Applications that use WSAGetOverlappedResult() (with the fWait flag set to false) in a polling mode to determine when an overlapped operation has completed will get this error code until the operation is complete.

WSA_IO_PENDING (OS dependent) Overlapped operations will complete later.

The application has initiated an overlapped operation which cannot be completed immediately. A completion indication will be given at a later time when the operation has been completed.

WSA_NOT_ENOUGH_MEMORY (OS dependent) Insufficient memory available.

Socket Error.txt

An application used a WinSock function which directly maps to a Win32 function. The Win32 function is indicating a lack of required memory resources.7