**select function**

The **select** function determines the status of one or more sockets, waiting if necessary, to perform synchronous I/O.

**Syntax**

C++

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_2822500b-a208-4614-8490-9e330babeb91');" \o "Copy to clipboard.)

int select(

\_In\_     int nfds,

\_Inout\_  fd\_set \*readfds,

\_Inout\_  fd\_set \*writefds,

\_Inout\_  fd\_set \*exceptfds,

\_In\_     const struct timeval \*timeout

);

**Parameters**

*nfds* [in]

Ignored. The *nfds* parameter is included only for compatibility with Berkeley sockets.

*readfds* [in, out]

An optional pointer to a set of sockets to be checked for readability.

*writefds* [in, out]

An optional pointer to a set of sockets to be checked for writability.

*exceptfds* [in, out]

An optional pointer to a set of sockets to be checked for errors.

*timeout* [in]

The maximum time for **select** to wait, provided in the form of a [**TIMEVAL**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740560(v=vs.85).aspx) structure. Set the *timeout* parameter to **null** for blocking operations.

**Return value**

The **select** function returns the total number of socket handles that are ready and contained in the [**fd\_set**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms737873(v=vs.85).aspx) structures, zero if the time limit expired, or SOCKET\_ERROR if an error occurred. If the return value is SOCKET\_ERROR, [**WSAGetLastError**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741580(v=vs.85).aspx) can be used to retrieve a specific error code.

|  |  |
| --- | --- |
| **Error code** | **Meaning** |
| [**WSANOTINITIALISED**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740668(v=vs.85).aspx#WSANOTINITIALISED) | A successful [**WSAStartup**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms742213(v=vs.85).aspx) call must occur before using this function. |
| [**WSAEFAULT**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740668(v=vs.85).aspx#WSAEFAULT) | The Windows Sockets implementation was unable to allocate needed resources for its internal operations, or the *readfds*, *writefds*, *exceptfds*, or *timeval* parameters are not part of the user address space. |
| [**WSAENETDOWN**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740668(v=vs.85).aspx#WSAENETDOWN) | The network subsystem has failed. |
| [**WSAEINVAL**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740668(v=vs.85).aspx#WSAEINVAL) | The *time-out* value is not valid, or all three descriptor parameters were **null**. |
| [**WSAEINTR**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740668(v=vs.85).aspx#WSAEINTR) | A blocking Windows Socket 1.1 call was canceled through [**WSACancelBlockingCall**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741547(v=vs.85).aspx). |
| [**WSAEINPROGRESS**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740668(v=vs.85).aspx#WSAEINPROGRESS) | A blocking Windows Sockets 1.1 call is in progress, or the service provider is still processing a callback function. |
| [**WSAENOTSOCK**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740668(v=vs.85).aspx#WSAENOTSOCK) | One of the descriptor sets contains an entry that is not a socket. |

**Remarks**

The **select** function is used to determine the status of one or more sockets. For each socket, the caller can request information on read, write, or error status. The set of sockets for which a given status is requested is indicated by an [**fd\_set**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms737873(v=vs.85).aspx) structure. The sockets contained within the **fd\_set** structures must be associated with a single service provider. For the purpose of this restriction, sockets are considered to be from the same service provider if the [**WSAPROTOCOL\_INFO**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741675(v=vs.85).aspx) structures describing their protocols have the same *providerId* value. Upon return, the structures are updated to reflect the subset of these sockets that meet the specified condition. The **select** function returns the number of sockets meeting the conditions. A set of macros is provided for manipulating an **fd\_set** structure. These macros are compatible with those used in the Berkeley software, but the underlying representation is completely different.

The parameter *readfds* identifies the sockets that are to be checked for readability. If the socket is currently in the [**listen**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms739168(v=vs.85).aspx) state, it will be marked as readable if an incoming connection request has been received such that an [**accept**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms737526(v=vs.85).aspx) is guaranteed to complete without blocking. For other sockets, readability means that queued data is available for reading such that a call to [**recv**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740121(v=vs.85).aspx), [**WSARecv**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741688(v=vs.85).aspx), [**WSARecvFrom**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741686(v=vs.85).aspx), or [**recvfrom**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740120(v=vs.85).aspx) is guaranteed not to block.

For connection-oriented sockets, readability can also indicate that a request to close the socket has been received from the peer. If the virtual circuit was closed gracefully, and all data was received, then a [**recv**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740121(v=vs.85).aspx) will return immediately with zero bytes read. If the virtual circuit was reset, then a **recv** will complete immediately with an error code such as [WSAECONNRESET](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740668(v=vs.85).aspx#wsaeconnreset). The presence of OOB data will be checked if the socket option SO\_OOBINLINE has been enabled (see [**setsockopt**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740476(v=vs.85).aspx)).

The parameter *writefds* identifies the sockets that are to be checked for writability. If a socket is processing a [**connect**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms737625(v=vs.85).aspx) call (nonblocking), a socket is writeable if the connection establishment successfully completes. If the socket is not processing a **connect** call, writability means a [**send**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740149(v=vs.85).aspx), [**sendto**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740148(v=vs.85).aspx), or [**WSASendto**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741693(v=vs.85).aspx) are guaranteed to succeed. However, they can block on a blocking socket if the *len* parameter exceeds the amount of outgoing system buffer space available. It is not specified how long these guarantees can be assumed to be valid, particularly in a multithreaded environment.

The parameter *exceptfds* identifies the sockets that are to be checked for the presence of OOB data or any exceptional error conditions.

**Note**  Out-of-band data will only be reported in this way if the option SO\_OOBINLINE is **FALSE**. If a socket is processing a [**connect**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms737625(v=vs.85).aspx) call (nonblocking), failure of the connect attempt is indicated in *exceptfds* (application must then call [**getsockopt**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms738544(v=vs.85).aspx) SO\_ERROR to determine the error value to describe why the failure occurred). This document does not define which other errors will be included.

Any two of the parameters, *readfds*, *writefds*, or *exceptfds*, can be given as **null**. At least one must be non-**null**, and any non-**null** descriptor set must contain at least one handle to a socket.

In summary, a socket will be identified in a particular set when **select** returns if:

***readfds*:**

* **If** [**listen**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms739168(v=vs.85).aspx) **has been called and a connection is pending,** [**accept**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms737526(v=vs.85).aspx) **will succeed.**
* **Data is available for reading (includes OOB data if SO\_OOBINLINE is enabled).**
* **Connection has been closed/reset/terminated.**

***writefds*:**

* **If processing a** [**connect**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms737625(v=vs.85).aspx) **call (nonblocking), connection has succeeded.**
* **Data can be sent.**

***exceptfds*:**

* **If processing a connect call (nonblocking), connection attempt failed.**
* **OOB data is available for reading (only if SO\_OOBINLINE is disabled).**

Four macros are defined in the header file Winsock2.h for manipulating and checking the descriptor sets. The variable FD\_SETSIZE determines the maximum number of descriptors in a set. (The default value of FD\_SETSIZE is 64, which can be modified by defining FD\_SETSIZE to another value before including Winsock2.h.) Internally, socket handles in an [**fd\_set**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms737873(v=vs.85).aspx) structure are not represented as bit flags as in Berkeley Unix. Their data representation is opaque. Use of these macros will maintain software portability between different socket environments. The macros to manipulate and check **fd\_set** contents are:

**FD\_CLR(***s***,** \**set***)**

Removes the descriptor *s* from *set*.

**FD\_ISSET(***s***,** \**set***)**

Nonzero if *s* is a member of the *set*. Otherwise, zero.

**FD\_SET(***s***,** \**set***)**

Adds descriptor *s* to *set*.

**FD\_ZERO(**\**set***)**

Initializes the *set* to the **null** set.

The parameter *time-out* controls how long the **select** can take to complete. If *time-out* is a **null** pointer, **select** will block indefinitely until at least one descriptor meets the specified criteria. Otherwise, *time-out* points to a [**TIMEVAL**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms740560(v=vs.85).aspx) structure that specifies the maximum time that **select** should wait before returning. When **select** returns, the contents of the **TIMEVAL** structure are not altered. If **TIMEVAL** is initialized to {0, 0}, **select** will return immediately; this is used to poll the state of the selected sockets. If **select** returns immediately, then the **select** call is considered nonblocking and the standard assumptions for nonblocking calls apply. For example, the blocking hook will not be called, and Windows Sockets will not yield.

**Note**  The **select** function has no effect on the persistence of socket events registered with [**WSAAsyncSelect**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741540(v=vs.85).aspx) or [**WSAEventSelect**](http://msdn.microsoft.com/en-us/library/windows/desktop/ms741576(v=vs.85).aspx).

**Note**  When issuing a blocking Winsock call such as **select** with the *timeout* parameter set to **NULL**, Winsock may need to wait for a network event before the call can complete. Winsock performs an alertable wait in this situation, which can be interrupted by an asynchronous procedure call (APC) scheduled on the same thread. Issuing another blocking Winsock call inside an APC that interrupted an ongoing blocking Winsock call on the same thread will lead to undefined behavior, and must never be attempted by Winsock clients.

选择功能  
select函数确定一个或多个套接字的状态，如果有必要等待，执行同步I / O。  
  
句法  
C + +的复制  
   
整型选择（  
  \_In\_ INT NFDs的，  
  \_Inout\_的fd\_set \* readfds ，  
  \_Inout\_的fd\_set \* writefds ，  
  \_Inout\_的fd\_set \* exceptfds中，  
  \_In\_常量timeval结构\*超时  
） ;  
  
参数  
NFDs的[中]  
忽略。该NFDs的参数只是为了与Berkeley套接字兼容。  
  
readfds [ IN，OUT ]  
一个可选的指向一组插座被检查可读性。  
  
writefds [ IN，OUT ]  
一个可选的指向一组插座进行检查可写的。  
  
exceptfds中[ IN，OUT ]  
一个可选的指向一组插座进行检查的错误。  
  
超时[中]  
的最大时间选择等，以timeval结构的形式提供。设置超时参数设置为null阻止操作。  
  
返回值  
select函数将返回套接字句柄已经准备就绪，包含在fd\_set中的结构，零，如果期限届满，或SOCKET\_ERROR ，如果发生错误的总数。如果返回值是SOCKET\_ERROR ，可以通过WSAGetLastError用来检索特定的错误代码。  
  
错误代码含义  
WSANOTINITIALISED成功地调用WSAStartup调用使用此功能前必须发生。  
   
WSAEFAULT Windows Sockets实现无法分配所需的资源用于其内部运作，或readfds ， writefds ， exceptfds中，或timeval的参数不是用户地址空间的一部分。  
   
WSAENETDOWN网络子系统失效。  
   
WSAEINVAL超时值无效，或所有三个描述符参数为空。  
   
WSAEINTR一个阻塞的Windows套接字1.1呼叫通过WSACancelBlockingCall被取消。  
   
WSAEINPROGRESS一个阻塞的Windows Sockets 1.1通话正在进行中，或者服务提供者仍在处理一个回调函数。  
   
WSAENOTSOCK的描述符一组包含一个条目是不是一个套接字。  
   
  
   
  
备注  
选择函数被用来确定一个或多个插座的状态。对于每一个插座，来电者可以请求信息的读，写或错误状态。该组为其中一个给定的状态被请求的插座是由一个fd\_set的结构来表示。该包含在fd\_set中结构内插座必须配单服务提供商相关联。对于这种限制的目的，插座被认为是来自同一个服务提供商如果WSAPROTOCOL\_INFO结构描述他们的协议具有相同的providerId值。返回时，该结构被更新，以反映这些插槽，满足给定条件的子集。 select函数返回符合条件的插槽数量。提供了一种操作fd\_set的结构一组宏。这些宏是与那些在伯克利软件兼容使用，但潜在的表示是完全不同的。  
  
参数readfds标识要被检查可读性的插座。如果套接字目前处于listen状态，它将被标记为可读，如果传入的连接请求已收到，这样一个accept保证不阻塞才能完成。对于其他插座，可读性意味着排队的数据可供读取，这样调用recv的，的WSARecv ， WSARecvFrom ，或recvfrom保证不会阻拦。  
  
对于面向连接的套接字，可读性也可以表明一个请求关闭套接字已接收到对等。如果虚电路被关闭摆好，并被接收的所有数据，那么recv的将立即返回读取零字节。如果虚电路被复位，则recv的将立即完成并返回错误代码，如​​WSAECONNRESET 。如果套接字选项SO\_OOBINLINE已启用（见setsockopt的） OOB数据的存在将被检查。  
  
参数writefds标识要检查可写性的插座。如果一个套接字处理一个连接调用（非阻塞） ，套接字是可写的，如果连接建立成功完成。如果套接字没有处理一个连接的呼叫，可写性意味着一个发送， SENDTO ，或WSASendto保证可以成功。然而，它们可以在阻塞套接字，如果len参数超过了即将离任的系统缓冲区可用空间量封锁。它不指定这些保证多久可以认为是有效的，特别是在多线程环境中。  
  
参数exceptfds中标识要检查OOB数据或任何特殊的错误情况存在的插座。  
  
注出带外数据将只报以这种方式，如果SO\_OOBINLINE选项为FALSE 。如果一个套接字处理一个连接调用（非阻塞） ，连接尝试失败是表示在exceptfds中（应用程序必须调用getsockopt函数SO\_ERROR来确定误差值来描述发生故障的原因） 。本文件并不确定哪些其他错误将被包括在内。  
  
任意两个参数， readfds ， writefds或exceptfds中的，可以表示为空。至少有一个必须为非空，非空描述符集必须至少包含一个手柄的插座。  
  
总之，一个插座将被确定在一组特定的，如果当select返回：  
  
readfds ：  
  
•如果听被称为和连接挂起，接受一定会成功。  
•数据可供读取（包括OOB数据，如果SO\_OOBINLINE已启用） 。  
•连接已经关闭/重置/终止。  
writefds ：  
  
•如果处理一个连接调用（非阻塞） ，连接成功。  
•数据可以被发送。  
exceptfds中：  
  
•如果处理一个连接调用（非阻塞） ，连接尝试失败。  
• OOB数据可用于读取（仅当SO\_OOBINLINE被禁用） 。  
四宏在头文件Winsock 2.h中定义的操作和检查描述符集。变量FD\_SETSIZE决定在一组描述符的最大数量。 （ FD\_SETSIZE的默认值是64 ，它可以通过定义FD\_SETSIZE为其他值，包括Winsock 2.h中之前进行修改。 ）内部，在FD\_SET结构socket句柄，并不表示为位标志在伯克利的Unix 。他们的数据表示是不透明的。使用这些宏将保持不同的插座环境之间软件的可移植性。操作和检查的fd\_set内容的宏有：  
  
  
FD\_CLR （ S， \*套）  
从集合中删除描述符s 。  
  
FD\_ISSET （ S， \*套）  
非零，如果s是集合的成员。否则为零。  
  
FD\_SET （ S， \*套）  
添加描述符s来设置。  
  
FD\_ZERO （ \*套）  
初始化设置为空集。  
  
参数超时控制多久选择可以完成。如果超时是一个空指针，选择将无限期地阻塞，直到至少有一个描述符符合规定的标准。否则，超时指向的timeval结构，指定选择最大的时间应在返回前等待。当选择的回报， timeval结构的内容不改变。如果timeval中被初始化为{ 0,0} ，选择将立即返回;这是用来轮询选定插座的状态。如果选择立即返回，然后选择呼叫被认为是非阻塞和非阻塞调用的标准假设适用。例如，阻塞钩子函数就不会被调用，而Windows套接字不会屈服。  
  
注意选择功能对用的WSAAsyncSelect或WSAEventSelect注册的套接字事件的持久性没有影响。  
  
注意：当发出一个阻塞的Winsock调用，如选择与设置为NULL timeout参数， Winsock的可能需要等待网络事件之前的调用可以完成。 Winsock的执行一个警惕的等待在这种情况下，可以通过安排在同一个线程异步过程调用（APC ）被打断。里面发出的APC另一个阻塞的Winsock调用，在同一个线程中断正在进行的封锁Winsock调用将导致不确定的行为，绝不能由Winsock的客户端尝试