

GetOverlappedResult function

Retrieves the results of an overlapped operation on the specified file, named pipe, or communications device. To specify a timeout interval or wait on an alertable thread, use [GetOverlappedResultEx](#).

Syntax

C++

```
BOOL WINAPI GetOverlappedResult(  
    _In_   HANDLE      hFile,  
    _In_   LPOVERLAPPED lpOverlapped,  
    _Out_  LPDWORD      lpNumberOfBytesTransferred,  
    _In_   BOOL        bWait  
);
```

Parameters

hFile [in]

A handle to the file, named pipe, or communications device. This is the same handle that was specified when the overlapped operation was started by a call to the [ReadFile](#), [WriteFile](#), [ConnectNamedPipe](#), [TransactNamedPipe](#), [DeviceIoControl](#), or [WaitCommEvent](#) function.

lpOverlapped [in]

A pointer to an **OVERLAPPED** structure that was specified when the overlapped operation was started.

lpNumberOfBytesTransferred [out]

A pointer to a variable that receives the number of bytes that were actually transferred by a read or write operation. For a [TransactNamedPipe](#) operation, this is the number of bytes that were read from the pipe. For a [DeviceIoControl](#) operation, this is the number of bytes of output data returned by the device driver. For a [ConnectNamedPipe](#) or [WaitCommEvent](#) operation, this value is undefined.

bWait [in]

If this parameter is **TRUE**, and the **Internal** member of the *lpOverlapped* structure is **STATUS_PENDING**, the function does not return until the operation has been completed. If this parameter is **FALSE** and the operation is still pending, the function returns **FALSE** and the [GetLastError](#) function returns **ERROR_IO_INCOMPLETE**.

Return value

If the function succeeds, the return value is nonzero.

If the function fails, the return value is zero. To get extended error information, call [GetLastError](#).

Remarks

The results reported by the **GetOverlappedResult** function are those of the specified handle's last overlapped operation to which the specified **OVERLAPPED** structure was provided, and for which the operation's results were pending. A pending operation is indicated when the function that started the operation returns **FALSE**, and the [GetLastError](#) function returns **ERROR_IO_PENDING**. When an I/O operation is pending, the function that started the operation resets the **hEvent** member of the **OVERLAPPED** structure to the nonsignaled state. Then when the pending operation has been completed, the system sets the event object to the signaled state.

If the *bWait* parameter is **TRUE**, **GetOverlappedResult** determines whether the pending operation has been completed by waiting for the event object to be in the signaled state.

If the **hEvent** member of the **OVERLAPPED** structure is **NULL**, the system uses the state of the *hFile* handle to signal when the operation has been completed. Use of file, named pipe, or communications-device handles for this purpose is discouraged. It is safer to use an event object because of the confusion that can occur when multiple simultaneous overlapped operations are performed on the same file, named pipe, or communications device. In this situation, there is no way to know which operation caused the object's state to be signaled.

Examples

For an example that uses **GetOverlappedResult**, see [Testing for the End of a File](#).

Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Minimum supported phone	Windows Phone 8
Header	WinBase.h on Windows XP, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008, and Windows Server 2008 R2 (include Windows.h); Ioapiset.h on Windows 8 and Windows Server 2012

Library	Kernel32.lib
DLL	Kernel32.dll

See also

[Cancello](#)

[ConnectNamedPipe](#)

[CreateEvent](#)

[DeviceIoControl](#)

[GetLastError](#)

[GetOverlappedResultEx](#)

[OVERLAPPED](#)

[Overlapped Input and Output](#)

[ReadFile](#)

[Synchronization Functions](#)

[TransactNamedPipe](#)

[WaitCommEvent](#)

[WriteFile](#)