

# SetEndOfFile function

Sets the physical file size for the specified file to the current position of the file pointer.

The physical file size is also referred to as the end of the file. The **SetEndOfFile** function can be used to truncate or extend a file. To set the logical end of a file, use the [SetFileValidData](#) function.

## Syntax

**C++**

```
BOOL WINAPI SetEndOfFile(  
    _In_ HANDLE hFile  
);
```

## Parameters

*hFile* [in]

A handle to the file to be extended or truncated.

The file handle must be created with the **GENERIC\_WRITE** access right. For more information, see [File Security and Access Rights](#).

## Return value

If the function succeeds, the return value is nonzero.

If the function fails, the return value is zero (0). To get extended error information, call [GetLastError](#).

## Remarks

The **SetEndOfFile** function can be used to truncate or extend a file. If the file is extended, the contents of the file between the old end of the file and the new end of the file are not defined.

Each file stream has the following:

- File size: the size of the data in a file, to the byte.
- Allocation size: the size of the space that is allocated for a file on a disk, which is always an even multiple of the cluster size.
- Valid data length: the length of the data in a file that is actually written, to the byte. This value is always less

than or equal to the file size.

The **SetEndOfFile** function sets the file size. Use [SetFileValidData](#) to set the valid data length.

If [CreateFileMapping](#) is called to create a file mapping object for *hFile*, [UnmapViewOfFile](#) must be called first to unmap all views and call [CloseHandle](#) to close the file mapping object before you can call **SetEndOfFile**.

### Transacted Operations

If there is a transaction bound to the handle, then the change in the end-of-file position is transacted.

In Windows 8 and Windows Server 2012, this function is supported by the following technologies.

| Technology                                | Supported |
|---|-----------|
| Server Message Block (SMB) 3.0 protocol   | Yes       |
| SMB 3.0 Transparent Failover (TFO)        | Yes       |
| SMB 3.0 with Scale-out File Shares (SO)   | Yes       |
| Cluster Shared Volume File System (CsvFS) | Yes       |
| Resilient File System (ReFS)              | Yes       |

## Requirements

|                                 |  |
|---------------------------------|--|
| <b>Minimum supported client</b> | Windows XP [desktop apps   Windows Store apps]   |
| <b>Minimum supported server</b> | Windows Server 2003 [desktop apps   Windows Store apps]  |
| <b>Minimum supported phone</b>  | Windows Phone 8  |
| <b>Header</b>                   | FileAPI.h (include Windows.h);<br>WinBase.h on Windows Server 2008 R2, Windows 7, Windows Server 2008, |

|                |  |
|----------------|--|
|                | Windows Vista, Windows Server 2003, and Windows XP (include Windows.h) |
| <b>Library</b> | Kernel32.lib   |
| <b>DLL</b>     | Kernel32.dll   |

## See also

[CloseHandle](#)

[CreateFile](#)

[CreateFileMapping](#)

[File Management Functions](#)

[SetFileInformationByHandle](#)

[SetFileValidData](#)

[UnmapViewOfFile](#)