OpenFile function (winbase.h)

Article10/12/2021

Creates, opens, reopens, or deletes a file.

> **Note** This function has limited capabilities and is not recommended. For new application development, use the **CreateFile** function.

Syntax

C++

```
HFILE OpenFile(
  [in]  LPCSTR     lpFileName,
  [out] LPOFSTRUCT lpReOpenBuff,
  [in]  UINT       uStyle
);
```

Parameters

`[in] lpFileName`

The name of the file.

The string must consist of characters from the 8-bit Windows character set. The **OpenFile** function does not support

Unicode file names or opening named pipes.

`[out] lpReOpenBuff`

A pointer to the OFSTRUCT structure that receives information about a file when it is first opened.

The structure can be used in subsequent calls to the **OpenFile** function to see an open file.

The OFSTRUCT structure contains a path string member with a length that is limited to **OFS_MAXPATHNAME** characters, which is 128 characters. Because of this, you cannot use the **OpenFile** function to open a file with a path length that exceeds 128 characters. The CreateFile function does not have this path length limitation.

`[in] uStyle`

The action to be taken.

This parameter can be one or more of the following values.

Expand table

Value	Meaning
OF_CANCEL 0x00000800	Ignored. To produce a dialog box containing a **Cancel** button, use **OF_PROMPT**.
OF_CREATE 0x00001000	Creates a new file. If the file exists, it is truncated to zero (0) length.
OF_DELETE 0x00000200	Deletes a file.
OF_EXIST 0x00004000	Opens a file and then closes it. Use this to test for the existence of a file.

OF_PARSE 0x00000100	Fills the OFSTRUCT structure, but does not do anything else.
OF_PROMPT 0x00002000	Displays a dialog box if a requested file does not exist. A dialog box informs a user that the system cannot find a file, and it contains **Retry** and **Cancel** buttons. The **Cancel** button directs **OpenFile** to return a file-not-found error message.
OF_READ 0x00000000	Opens a file for reading only.
OF_READWRITE 0x00000002	Opens a file with read/write permissions.
OF_REOPEN 0x00008000	Opens a file by using information in the reopen buffer.
OF_SHARE_COMPAT 0x00000000	For MS-DOS–based file systems, opens a file with compatibility mode, allows any process on a specified computer to open the file any number of times. Other efforts to open a file with other sharing modes fail. This flag is mapped to the **FILE_SHARE_READ|FILE_SHARE_WRITE** flags of the CreateFile function.
OF_SHARE_DENY_NONE 0x00000040	Opens a file without denying read or write access to other processes. On MS-DOS-based file systems, if the file has been opened in compatibility mode by any other process, the function fails. This flag is mapped to the **FILE_SHARE_READ|FILE_SHARE_WRITE** flags of the CreateFile function.
OF_SHARE_DENY_READ 0x00000030	Opens a file and denies read access to other processes. On MS-DOS-based file systems, if the file has been opened in compatibility mode, or for read access by any other process, the function fails. This flag is mapped to the **FILE_SHARE_WRITE** flag of the CreateFile function.

OF_SHARE_DENY_WRITE 0x00000020	Opens a file and denies write access to other processes. On MS-DOS-based file systems, if a file has been opened in compatibility mode, or for write access by any other process, the function fails. This flag is mapped to the **FILE_SHARE_READ** flag of the CreateFile function.
OF_SHARE_EXCLUSIVE 0x00000010	Opens a file with exclusive mode, and denies both read/write access to other processes. If a file has been opened in any other mode for read/write access, even by the current process, the function fails.
OF_VERIFY	Verifies that the date and time of a file are the same as when it was opened previously. This is useful as an extra check for read-only files.
OF_WRITE 0x00000001	Opens a file for write access only.

Return value

If the function succeeds, the return value specifies a file handle to use when performing file I/O. To close the file, call the CloseHandle function using this handle.

If the function fails, the return value is **HFILE_ERROR**. To get extended error information, call GetLastError.

Remarks

If the *lpFileName* parameter specifies a file name and extension only, this function searches for a matching file in the following directories and the order shown:

1. The directory where an application is loaded.

2. The current directory.
3. The Windows system directory.

   Use the GetSystemDirectory function to get the path of this directory.
4. The 16-bit Windows system directory.

   There is not a function that retrieves the path of this directory, but it is searched.
5. The Windows directory.

   Use the GetWindowsDirectory function to get the path of this directory.
6. The directories that are listed in the PATH environment variable.

The *lpFileName* parameter cannot contain wildcard characters.

The **OpenFile** function does not support the **OF_SEARCH** flag that the 16-bit Windows **OpenFile** function supports. The **OF_SEARCH** flag directs the system to search for a matching file even when a file name includes a full path. Use the SearchPath function to search for a file.

A sharing violation occurs if an attempt is made to open a file or directory for deletion on a remote machine when the value of the *uStyle* parameter is the **OF_DELETE** access flag OR'ed with any other access flag, and the remote file or directory has not been opened with **FILE_SHARE_DELETE** share access. To avoid the sharing violation in this scenario, open the remote file or directory with **OF_DELETE** access only, or call DeleteFile without first opening the file or directory for deletion.

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

Expand table

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

CsvFs will do redirected IO for compressed files.

Requirements

Expand table

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Target Platform	Windows
Header	winbase.h (include Windows.h)
Library	Kernel32.lib
DLL	Kernel32.dll

See also

CreateFile

File Management Functions

GetSystemDirectory

GetWindowsDirectory

OFSTRUCT

SearchPath

Feedback

Was this page helpful? Yes No