**Opening Files**

In MFC, the most common way to open a file is a two-stage process.

**To open a file**

1. Create the file object without specifying a path or permission flags.

You usually create a file object by declaring a [CFile](http://msdn.microsoft.com/en-us/library/60fh2b6f.aspx) variable on the stack frame.

1. Call the [Open](http://msdn.microsoft.com/en-us/library/hwbccf8z.aspx) member function for the file object, supplying a path and permission flags.

**The return value for Open will be nonzero if the file was opened successfully or 0 if the specified file could not be opened**. The **Open** member function is prototyped as follows:

virtual BOOL Open( LPCTSTR lpszFileName, UINT nOpenFlags, CFileException\* pError = NULL );

The open flags specify which permissions, such as read-only, you want for the file. The possible flag values are defined as enumerated constants within the **CFile** class, so they are qualified with "**CFile::**" as in **CFile::modeRead**. Use the **CFile::modeCreate** flag if you want to create the file.

The following example shows how to create a new file with read/write permission (replacing any previous file with the same path):

TCHAR\* pszFileName = \_T("c:\\test\\myfile.dat");

CFile myFile;

CFileException fileException;

if ( !myFile.Open( pszFileName, CFile::modeCreate |

CFile::modeReadWrite, &fileException ) )

{

TRACE( \_T("Can't open file %s, error = %u\n"),

pszFileName, fileException.m\_cause );

}

# Reading and Writing Files

If you've used the C run-time library file-handling functions, MFC reading and writing operations will appear familiar. This article describes reading directly from and writing directly to a **CFile** object. You can also do buffered file I/O with the [CArchive](http://msdn.microsoft.com/en-us/library/caz3zy5s.aspx) class.

### To read from and write to the file

1. Use the **Read** and **Write** member functions to read and write data in the file.
2. The **Seek** member function is also available for moving to a specific offset within the file.

**Read** takes a pointer to a buffer and the number of bytes to read and returns the actual number of bytes that were read. If the required number of bytes could not be read because end-of-file (EOF) is reached, the actual number of bytes read is returned. If any read error occurs, an exception is thrown. **Write**is similar to **Read**, but the number of bytes written is not returned. If a write error occurs, including not writing all the bytes specified, an exception is thrown. If you have a valid **CFile** object, you can read from it or write to it as shown in the following example:

TCHAR szBuffer[256];

UINT nActual = 0;

CFile myFile;

if ( myFile.Open( \_T("c:\\test\\myfile.dat"), CFile::modeCreate |

CFile::modeReadWrite ) )

{

myFile.Write( szBuffer, sizeof( szBuffer ) );

myFile.Flush();

myFile.Seek( 0, CFile::begin );

nActual = myFile.Read( szBuffer, sizeof( szBuffer ) );

}

# Closing Files

As usual in I/O operations, once you finish with a file, you must close it.

### To close a file

* Use the **Close** member function. This function closes the file-system file and flushes buffers if necessary.

If you allocated the [CFile](http://msdn.microsoft.com/en-us/library/60fh2b6f.aspx) object on the frame (as in the example shown in [Opening Files](http://msdn.microsoft.com/en-us/library/fthw9e38.aspx)), the object will automatically be closed and then destroyed when it goes out of scope. Note that deleting the **CFile** object does not delete the physical file in the file system.

# Accessing File Status

**CFile** also supports getting file status, including whether the file exists, creation and modification dates and times, logical size, and path.

### To get file status

* Use the [CFile](http://msdn.microsoft.com/en-us/library/60fh2b6f.aspx) class to get and set information about a file. One useful application is to use the **CFile** static member function **GetStatus** to determine if a file exists.**GetStatus** returns 0 if the specified file does not exist.

Thus, you could use the result of **GetStatus** to determine whether to use the **CFile::modeCreate** flag when opening a file, as shown by the following example:

CFile theFile;

TCHAR\* szFileName = \_T("c:\\test\\myfile.dat");

BOOL bOpenOK;

CFileStatus status;

if( CFile::GetStatus( szFileName, status ) )

{

// Open the file without the Create flag

bOpenOK = theFile.Open( szFileName,

CFile::modeWrite );

}

else

{

// Open the file with the Create flag

bOpenOK = theFile.Open( szFileName,

CFile::modeCreate | CFile::modeWrite );

}

For related information, see [Serialization](http://msdn.microsoft.com/en-us/library/6bz744w8.aspx).

# CFile Members

**Data Members**

|  |  |
| --- | --- |
| **[m\_hFile](http://msdn.microsoft.com/en-US/library/f0wca4x2.aspx)** | Usually contains the operating-system file handle. |
| [CFile::hFileNull](http://msdn.microsoft.com/en-US/library/e1y65t41.aspx) | Determines if the **CFile** object has a valid handle. |
| [m\_pTM](http://msdn.microsoft.com/en-US/library/ee231006.aspx) | Pointer to CAtlTransactionManager object. |

**Operators**

|  |  |
| --- | --- |
| [operator HANDLE](http://msdn.microsoft.com/en-US/library/c7bcxbkx.aspx) | A handle to a **CFile** object. |

**Construction**

|  |  |
| --- | --- |
| [Abort](http://msdn.microsoft.com/en-US/library/015dc6cy.aspx) | Closes a file ignoring all warnings and errors. |
| [**CFile**](http://msdn.microsoft.com/en-US/library/cz0a83sb.aspx) | Constructs a **CFile** object from a path or file handle. |
| [**Close**](http://msdn.microsoft.com/en-US/library/sb6ka6t4.aspx) | Closes a file and deletes the object. |
| [Duplicate](http://msdn.microsoft.com/en-US/library/w9xtw1hy.aspx) | Constructs a duplicate object based on this file. |
| [**Open**](http://msdn.microsoft.com/en-US/library/hwbccf8z.aspx) | Safely opens a file with an error-testing option. |

**Input/Output**

|  |  |
| --- | --- |
| [**Flush**](http://msdn.microsoft.com/en-US/library/4d9chxf7.aspx) | Flushes any data yet to be written. |
| [**Read**](http://msdn.microsoft.com/en-US/library/ctka0kks.aspx) | Reads (unbuffered) data from a file at the current file position. |
| [**Write**](http://msdn.microsoft.com/en-US/library/esb6sz20.aspx) | Writes (unbuffered) data in a file to the current file position. |

**Position**

|  |  |
| --- | --- |
| [GetLength](http://msdn.microsoft.com/en-US/library/b569d0t4.aspx) | Retrieves the length of the file. |
| [**Seek**](http://msdn.microsoft.com/en-US/library/8c5ccz0x.aspx) | Positions the current file pointer. |
| [SeekToBegin](http://msdn.microsoft.com/en-US/library/2291by8s.aspx) | Positions the current file pointer at the beginning of the file. |
| [SeekToEnd](http://msdn.microsoft.com/en-US/library/t68z1bx0.aspx) | Positions the current file pointer at the end of the file. |
| [SetLength](http://msdn.microsoft.com/en-US/library/069dywd4.aspx) | Changes the length of the file. |

**Locking**

|  |  |
| --- | --- |
| [LockRange](http://msdn.microsoft.com/en-US/library/yb9xb7ef.aspx) | Locks a range of bytes in a file. |
| [UnlockRange](http://msdn.microsoft.com/en-US/library/s3da7xss.aspx) | Unlocks a range of bytes in a file. |

**Status**

|  |  |
| --- | --- |
| [GetFileName](http://msdn.microsoft.com/en-US/library/7ykxys19.aspx) | Retrieves the filename of the selected file. |
| [GetFilePath](http://msdn.microsoft.com/en-US/library/55bbx41s.aspx) | Retrieves the full file path of the selected file. |
| [GetFileTitle](http://msdn.microsoft.com/en-US/library/kycxa4kz.aspx) | Retrieves the title of the selected file. |
| [**GetPosition**](http://msdn.microsoft.com/en-US/library/6b2hf282.aspx) | Retrieves the current file pointer. |
| [**GetStatus**](http://msdn.microsoft.com/en-US/library/fa0hc0ht.aspx) | Retrieves the status of this open file. |
| [SetFilePath](http://msdn.microsoft.com/en-US/library/3fst6b25.aspx) | Sets the full file path of the selected file. |

**Static**

|  |  |
| --- | --- |
| [**GetStatus**](http://msdn.microsoft.com/en-US/library/fa0hc0ht.aspx) | Retrieves the status of the specified file (static, virtual function). |
| [**Remove**](http://msdn.microsoft.com/en-US/library/a9skzxhx.aspx) | Deletes the specified file (static function). |
| [**Rename**](http://msdn.microsoft.com/en-US/library/0y89a13f.aspx) | Renames the specified file (static function). |
| [**SetStatus**](http://msdn.microsoft.com/en-US/library/a6ce0788.aspx) | Sets the status of the specified file (static, virtual function). |