CopyFile function

Copies an existing file to a new file.

The CopyFileEx function provides two additional capabilities. **CopyFileEx** can call a specified callback function each time a portion of the copy operation is completed, and **CopyFileEx** can be canceled during the copy operation.

To perform this operation as a transacted operation, use the CopyFileTransacted function.

Syntax

C++

```
BOOL WINAPI CopyFile(
  _In_ LPCTSTR lpExistingFileName,
  _In_ LPCTSTR lpNewFileName,
  _In_ BOOL    bFailIfExists
);
```

Parameters

lpExistingFileName [in]

The name of an existing file.

In the ANSI version of this function, the name is limited to **MAX_PATH** characters. To extend this limit to 32,767 wide characters, call the Unicode version of the function and prepend "\\?\" to the path. For more information, see Naming a File.

If *lpExistingFileName* does not exist, **CopyFile** fails, and GetLastError returns **ERROR_FILE_NOT_FOUND**.

lpNewFileName [in]

The name of the new file.

In the ANSI version of this function, the name is limited to **MAX_PATH** characters. To extend this limit to 32,767 wide characters, call the Unicode version of the function and prepend "\\?\" to the path. For more information, see Naming a File.

bFailIfExists [in]

If this parameter is **TRUE** and the new file specified by *lpNewFileName* already exists, the function fails. If this parameter is **FALSE** and the new file already exists, the function overwrites the existing file and succeeds.

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 security resource properties (**ATTRIBUTE_SECURITY_INFORMATION**) for the existing file are copied to the new file.

Windows 7, Windows Server 2008 R2, Windows Server 2008, Windows Vista, Windows Server 2003, and Windows XP: Security resource properties for the existing file are not copied to the new file until Windows 8 and Windows Server 2012.

File attributes for the existing file are copied to the new file. For example, if an existing file has the **FILE_ATTRIBUTE_READONLY** file attribute, a copy created through a call to **CopyFile** will also have the **FILE_ATTRIBUTE_READONLY** file attribute. For more information, see Retrieving and Changing File Attributes.

This function fails with **ERROR_ACCESS_DENIED** if the destination file already exists and has the **FILE_ATTRIBUTE_HIDDEN** or **FILE_ATTRIBUTE_READONLY** attribute set.

When **CopyFile** is used to copy an encrypted file, it attempts to encrypt the destination file with the keys used in the encryption of the source file. If this cannot be done, this function attempts to encrypt the destination file with default keys. If neither of these methods can be done, **CopyFile** fails with an **ERROR_ENCRYPTION_FAILED** error code.

Symbolic link behavior—If the source file is a symbolic link, the actual file copied is the target of the symbolic link.

If the destination file already exists and is a symbolic link, the target of the symbolic link is overwritten by the source file.

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

Examples

For an example, see Retrieving and Changing File Attributes.

Requirements

Minimum supported client	Windows XP [desktop apps only]
Minimum supported server	Windows Server 2003 [desktop apps only]
Header	WinBase.h (include Windows.h)
Library	Kernel32.lib
DLL	Kernel32.dll
Unicode and ANSI names	**CopyFileW** (Unicode) and **CopyFileA** (ANSI)

See also

CopyFileEx
CopyFileTransacted
CreateFile
File Attribute Constants
File Management Functions
MoveFile
Symbolic Links