

File Streams (Local File Systems) - Win32 apps

By jwmsft

Archived: 2026-04-05 22:42:25 UTC

A stream is a sequence of bytes. In the NTFS file system, streams contain the data that is written to a file, and that gives more information about a file than attributes and properties. For example, you can create a stream that contains search keywords, or the identity of the user account that creates a file.

Each stream that is associated with a file has its own allocation size, actual size, and valid data length:

- The allocation size is the amount of disk space that is reserved for a stream.
- The actual size is the number of bytes that are being used by a caller.
- The valid data length (VDL) is the number of bytes that are initialized from the allocation size for the stream.

Each stream also maintains its own state for compression, encryption, and sparseness. The **FILE_ATTRIBUTE_SPARSE_FILE** attribute on the file is set in the **dwFileAttributes** member of the [WIN32_FIND_DATA](#) structure returned from the [FindFirstFile](#), [FindFirstFileEx](#), and [FindNextFile](#) functions if any of the streams have ever been sparse. [GetFileAttributes](#), [GetFileAttributesEx](#), [GetFileAttributesTransacted](#), [GetFileInformationByHandle](#), and [GetFileInformationByHandleEx](#) return the sparse state of the default data stream if no stream is specified.

There are no file times associated with a stream. The file times for a file are updated when any stream in a file is updated.

Opportunistic locks are maintained per stream. Sharing modes are also maintained per stream. When delete access is requested on a file, the operating system checks for delete access on all open streams in a file. If another process has opened a stream without the **FILE_SHARE_DELETE** permission, you cannot open the file for delete access.

If a file being copied has a data stream and the network redirector is used, the file can only be copied if the client has both the read permission and the read attributes permission.

When specified from the Windows shell command line, the full name of a stream is "*filename:stream name:stream type*", as in the following example: "myfile.dat:stream1:\$DATA".

Any characters that are legal for a file name are also legal for the stream name, including spaces. For more information, see [Naming a File](#). The stream type (also called an attribute type code) is internal to the NTFS file system. Users therefore can't create new stream types, but they can open existing NTFS file system types. Stream type specifier values always start with the dollar sign (\$) symbol. See below for a list of stream types.

By default, the default data stream is unnamed. To fully specify the default data stream, use "*filename::\$DATA*", where \$DATA is the stream type. This is the equivalent of "*filename*". You can create a named stream in the file using the [file naming conventions](#). Note that "\$DATA" is a legal stream name. For example, the full name of a

stream named "\$DATA" on a file named "sample" would be "sample:\$DATA:\$DATA". If you created a stream named "bar" on the same file its full name would be "sample:bar:\$DATA".

When creating and working with files that have one-character names, prefix the file name with period followed by a backslash (.) or use a fully qualified path name. The reason to do this is that Windows treats one-character file names as drive letters. When a drive letter is specified with a relative path, a colon separates the drive letter from the path. When there is an ambiguity about whether a one-character name is a drive letter or a file name, Windows assumes it is a drive letter if the string following the colon is a valid path, even if the drive letter is invalid.

Following is the list of NTFS stream types, also called attribute type codes. Some of the stream types are internal to NTFS and their format is undocumented.

Stream Type	Description
::\$ATTRIBUTE_LIST	Contains a list of all attributes that make up the file and identifies where each attribute is located.
::\$BITMAP	A bitmap used by indexes to manage the b-tree free space for a directory. The b-tree is managed in 4 KB chunks (regardless of cluster size) and this is used to manage the allocation of these chunks. This stream type is present on every directory.
::\$DATA	Data stream. The default data stream has no name. Data streams can be enumerated using the FindFirstStreamW and FindNextStreamW functions.
::\$EA	Contains Extended Attributes data.
::\$EA_INFORMATION	Contains support information about the Extended Attributes.
::\$FILE_NAME	The name of the file, in Unicode characters. This includes the short name of the file as well as any hard links.
::\$INDEX_ALLOCATION	The stream type of a directory. Used to implement filename allocation for large directories. This stream represents the directory itself and contains all of the data of the directory. Changes to streams of this type are logged to the NTFS change journal. The default stream name of an \$INDEX_ALLOCATION stream type is \$I30 so "DirName", "DirName::\$INDEX_ALLOCATION", and "DirName:\$I30:\$INDEX_ALLOCATION" are all equivalent.
::\$INDEX_ROOT	This stream represents root of the b-tree of an index. This stream type is present on every directory.
::\$LOGGED_UTILTY_STREAM	Similar to::\$DATA but operations are logged to the NTFS change journal. Used by EFS and Transactional NTFS (TxF) . The ":StreamName:\$StreamType" pair for EFS is

Stream Type	Description
	":\$EFS:\$LOGGED_UTILITY_STREAM" and for TxF is "\$TXF_DATA:\$LOGGED_UTILITY_STREAM".
::\$OBJECT_ID	An 16-byte ID used to identify the file for the link-tracking service.
::\$REPARSE_POINT	The reparse point data.

[Using Streams](#)

[File naming conventions](#)

[Transactional NTFS \(TxF\)](#)

Source: <http://msdn.microsoft.com/en-us/library/aa364404>