

C# - Stream

C# includes following standard IO (Input/Output) classes to read/write from different sources like files, memory, network, isolated storage, etc.

**Stream:** *System.IO.Stream* is an abstract class that provides standard methods to transfer bytes (read, write, etc.) to the source. It is like a wrapper class to transfer bytes. Classes that need to read/write bytes from a particular source must implement the Stream class.

The following classes inherit Stream class to provide the functionality to Read/Write bytes from a particular source:

**FileStream** reads or writes bytes from/to a physical file, whether it is a .txt, .exe, .jpg, or any other file. FileStream is derived from the Stream class.

**MemoryStream:** MemoryStream reads or writes bytes that are stored in memory.

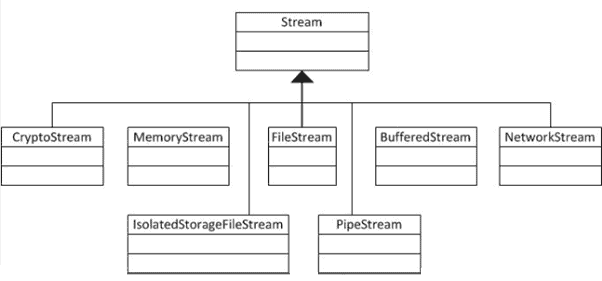
**BufferedStream:** BufferedStream reads or writes bytes from other Streams to improve certain I/O operations' performance.

**NetworkStream:** NetworkStream reads or writes bytes from a network socket.

**PipeStream:** PipeStream reads or writes bytes from different processes.

**CryptoStream:** CryptoStream is for linking data streams to cryptographic transformations.

The following diagram shows the hierarchy of stream classes:

[](https://www.tutorialsteacher.com/Content/images/csharp/stream-heirarchy.png)Stream Classes Hierarchy

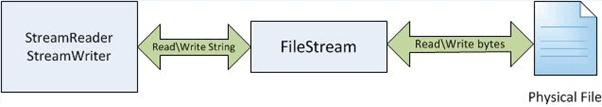
Stream Readers and Writers

**StreamReader**: StreamReader is a helper class for reading characters from a Stream by converting bytes into characters using an encoded value. It can be used to read strings (characters) from different Streams like FileStream, MemoryStream, etc.

**StreamWriter**: StreamWriter is a helper class for writing a string to a Stream by converting characters into bytes. It can be used to write strings to different Streams such as FileStream, MemoryStream, etc.

**BinaryReader**: BinaryReader is a helper class for reading primitive datatype from bytes.

**BinaryWriter**: BinaryWriter writes primitive types in binary.

[](https://www.tutorialsteacher.com/Content/images/csharp/stream-relations.png)Stream IO

The above image shows that FileStream reads bytes from a physical file, and then StreamReader reads strings by converting those bytes to strings. In the same way, the StreamWriter takes a string and converts it into bytes and writes to the FileStream, and then the FileStream writes the bytes to a physical file. So, the FileStream deals with bytes, where as StreamReader and StreamWriter deals with strings.

https://www.tutorialsteacher.com/Content/images/bulb-glow.png Points to Remember :

1. Stream is an abstract class for transfering bytes from different sources. It is base class for all other class that reads\writes bytes to different sources.
2. FileStream class provides reading and writing functionality of bytes to physical file.
3. Reader & writer classes provides functionality to read bytes from Stream classes (FileStream, MemoryStream etc) and converts bytes into appropriate encoding.
4. StreamReader provides a helper method to read string from FileStream by converting bytes into strings. StreamWriter provides a helper method to write string to FileStream by converting strings into bytes.

# How to write file using StreamWriter in C#?

//Create object of FileInfo for specified path

FileInfo fi = new FileInfo(@"D:\DummyFile.txt");

//Open file for Read\Write

FileStream fs = fi.Open(FileMode.OpenOrCreate, FileAccess.Write, FileShare.Read );

//Create StreamWriter object to write string to FileSream

StreamWriter sw = new StreamWriter(fs);

sw.WriteLine("Another line from streamwriter");

sw.Close();

Use the StreamReader class to read a physical file in C#. The following example shows how to read a file using StreamReader.

Example: Read a File using StreamReader

 Copy

//Create an object of FileInfo for specified path

FileInfo fi = new FileInfo(@"D:\DummyFile.txt");

//Open a file for Read\Write

FileStream fs = fi.Open(FileMode.OpenOrCreate, FileAccess.Read , FileShare.Read);

//Create an object of StreamReader by passing FileStream object on which it needs to operates on

StreamReader sr = new StreamReader(fs);

//Use the ReadToEnd method to read all the content from file

string fileContent = sr.ReadToEnd();

//Close the StreamReader object after operation

sr.Close();

fs.Close();

Notice that fi.Open() has three parameters: the first param is FileMode, used for creating a new file and opening it; the second parameter, FileAccess, is used to indicate a Read operation; and the third parameter is used to share the file with other users for reading purpose, while the file is open