File Handling In C#

In this article, you'll learn how to use files and folders related classes availalbe in .NET using C#.

Stream

When you open a file for reading or writing, it becomes stream. Stream is a sequence of bytes traveling from a source to a destination over a communication path.

The two basic streams are input and output streams. Input stream is used to read and output stream is used to write.

The System.IO namespace includes various classes for file handling.

The parent class of file processing is stream. Stream is an abstract class, which is used as the parent of the classes that actually implement the necessary operations.

The primary support of a file as an object is provided by a .NET Framework class called File. This static class is equipped with various types of (static) methods to create, save, open, copy, move, delete, or check the existence of a file.

Diagram to represent file-handling class hierarchy

Diagram

Description automatically generated

**Note**FileIno, DirectoryInfo and DriveInfo classes have instance methods. File, Directory, Path classes have static methods.

The following table describes some commonly used classes in the System.IO namespace.

|  |  |
| --- | --- |
| Class Name | Description |
| FileStream | It is used to read from and write to any location within a file |
| BinaryReader | It is used to read primitive data types from a binary stream |
| BinaryWriter | It is used to write primitive data types in binary format |
| StreamReader | It is used to read characters from a byte Stream |
| StreamWriter | It is used to write characters to a stream. |
| StringReader | It is used to read from a string buffer |
| StringWriter | It is used to write into a string buffer |
| DirectoryInfo | It is used to perform operations on directories |
| FileInfo | It is used to perform operations on files |

Reading and writing in the text file

**StreamWriter Class**

The StreamWriter class in inherited from the abstract class TextWriter. The TextWriter class represents a writer, which can write a series of characters.

The following table describes some of the methods used by StreamWriter class.

|  |  |
| --- | --- |
| Methods | Description |
| Close | Closes the current StreamWriter object and the underlying stream |
| Flush | Clears all buffers for the current writer and causes any buffered data to be written to the underlying stream |
| Write | Writes to the stream |
| WriteLine | Writes data specified by the overloaded parameters, followed by end of line |

**Program to write user input to a file using StreamWriter Class**

**using** System;

**using** System.Text;

**using** System.IO;

**namespace** FileWriting\_SW

{

**class** Program

    {

**class** FileWrite

        {

**public** **void** WriteData()

            {

                FileStream fs = **new** FileStream("c:\\test.txt", FileMode.Append, FileAccess.Write);

                StreamWriter sw = **new** StreamWriter(fs);

                Console.WriteLine("Enter the text which you want to write to the file");

**string** str = Console.ReadLine();

                sw.WriteLine(str);

                sw.Flush();

                sw.Close();

                fs.Close();

            }

        }

**static** **void** Main(**string**[] args)

        {

            FileWrite wr = **new** FileWrite();

            wr.WriteData();

        }

    }

}

StreamReader Class

The StreamReader class is inherited from the abstract class TextReader. The TextReader class represents a reader, which can read series of characters.

The following table describes some methods of the StreamReader class.

|  |  |
| --- | --- |
| Methods | Description |
| Close | Closes the object of StreamReader class and the underlying stream, and release any system resources associated with the reader |
| Peek | Returns the next available character but doesn't consume it |
| Read | Reads the next character or the next set of characters from the stream |
| ReadLine | Reads a line of characters from the current stream and returns data as a string |
| Seek | Allows the read/write position to be moved to any position with the file |

**Program to read from a file using StreamReader Class**

**using** System;

**using** System.IO;

**namespace** FileReading\_SR

{

**class** Program

    {

**class** FileRead

        {

**public** **void** ReadData()

            {

                FileStream fs = **new** FileStream("c:\\test.txt", FileMode.Open, FileAccess.Read);

                StreamReader sr = **new** StreamReader(fs);

                Console.WriteLine("Program to show content of test file");

                sr.BaseStream.Seek(0, SeekOrigin.Begin);

**string** str = sr.ReadLine();

**while** (str != **null**)

                {

                    Console.WriteLine(str);

                    str = sr.ReadLine();

                }

                Console.ReadLine();

                sr.Close();

                fs.Close();

            }

        }

**static** **void** Main(**string**[] args)

        {

            FileRead wr = **new** FileRead();

            wr.ReadData();

        }

    }

}

I hope that this article would have helped you in understanding file handling.