Telerik Academy

Text Files

Reading and Writing Text Files

C# Advanced

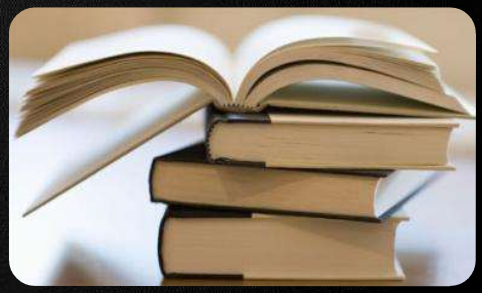
Telerik Software Academy

https://telerikacademy.com

Telerik Academy

Table of Contents

- What is Stream?
  - Stream Basics
- Reading Text Files
  - The `StreamReader` Class
- Writing Text Files
  - The `StreamWriter` Class
- Handling I/O Exceptions



Telerik Academy

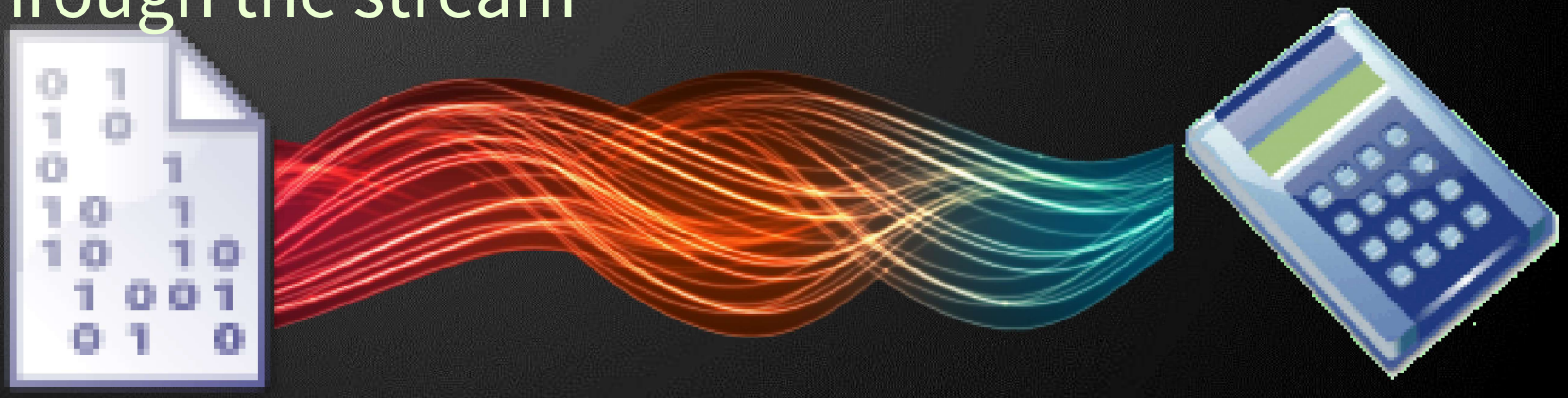
What Is Stream?

Streams Basic Concepts

Telerik Academy

What is Stream?

- Stream is the natural way to transfer data in the computer world
- To read or write a file, we open a stream connected to the file and access the data through the stream



Telerik Academy

Streams Basics

- Streams are used for reading and writing data into and from devices
- Streams are ordered sequences of bytes
  - Provide consecutive access to its elements
- Different types of streams are available to access different data sources:
  - File access, network access, memory streams and others
- Streams are open before using them and closed after that

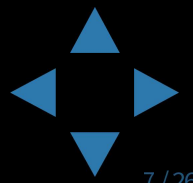
Telerik Academy

Reading Text Files

Telerik Academy

The StreamReader Class

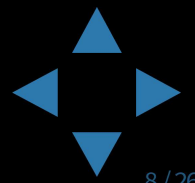
- System.IO.StreamReader
  - The easiest way to read a text file
  - Implements methods for reading text lines and sequences of characters
  - Constructed by file name or other stream
    - Can specify the text encoding
    - For Cyrillic use UTF8
  - Works like Console.Read() / ReadLine() but over text files



Telerik Academy

StreamReader Methods

- `new StreamReader(fileName)`
  - Constructor for creating reader from given file
- `StreamReader.ReadLine()`
  - Reads a single text line from the stream
  - Returns `null` when end-of-file is reached
- `StreamReader.ReadToEnd()`
  - Reads all the text until the end of the stream
- `StreamReader.Close()`
  - Closes the stream reader



Telerik Academy

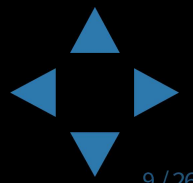
Reading a Text File

- Reading a text file and printing its content to the console:

```
StreamReader reader = new StreamReader("test.txt");
string fileContents = reader.ReadToEnd();
Console.WriteLine(fileContents);
streamReader.Close();
```

- Specifying the text encoding:

```
StreamReader reader = new StreamReader(
    "file-with-cyrillic.txt", Encoding.UTF8;
// Read the file contents here ...
reader.Close();
```

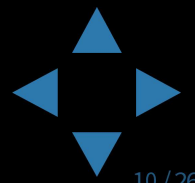
Telerik Academy

Using StreamReader – Practices

- The `StreamReader` instances should always be closed by calling the `Close()` method
  - Otherwise system resources can be lost
- In C# the preferable way to close streams and readers is by the `using` construction:

```
using (<stream object>)
{
   // Use the stream here. It will be closed at the end
}
```

- It automatically calls the `Close()` after the using construction is completed



Reading a Text File – *Example*

- Read and display a text file line by line:

```
string filename = "somefile.txt";
StreamReader reader = new StreamReader(filename);
using (reader)
{
    int lineNumber = 0;
    string line = reader.ReadLine();
    while (line != null)
    {
        lineNumber++;
        Console.WriteLine("Line {0}: {1}",
                          lineNumber, line);
        line = reader.ReadLine();
    }
}
```

Telerik Academy

Reading Text Files

Demo

Telerik Academy

Writing Text Files

Using the StreamWriter Class

Telerik Academy

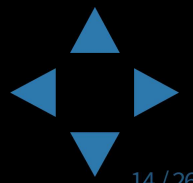
The StreamWriter Class

- System.IO.StreamWriter
  - Similar to StreamReader, but instead of reading, it provides writing functionality
- Constructed by file name or other stream

```
StreamWriter streamWriter = new StreamWriter(filename);
```

- Can define encoding
  - For Cyrillic use UTF8:

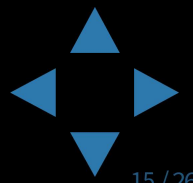
```
StreamWriter streamWriter =
    new StreamWriter(filename, false, Encoding.UTF8);
```



Telerik Academy

StreamWriter Methods

- `StreamWriter.Write()`
  - Writes string or other object to the stream
  - Like `Console.Write()`
- `StreamWriter.WriteLine()`
  - Like `Console.WriteLine()`
- `StreamWriter.Flush()`
  - Flushes the internal buffers to the hard drive
    - Or the stream
- `StreamWriter.AutoFlush`
  - Flush the internal buffer after each writing



Writing to a Text File – *Example*

- *Exampel:* Create text file named "numbers.txt" and print in it the numbers from 1 to 20 (one per line):

```
StreamWriter streamWriter =
    new StreamWriter("numbers.txt");
using (streamWriter)
{
    for (int number = 1; number <= 20; number++)
    {
        streamWriter.WriteLine(number);
    }
}
```

Telerik Academy

Writing Text Files

Demo

Telerik Academy

Reading and Writing Text Files: *Examples*

Telerik Academy

Counting Word Occurrences – *Example*

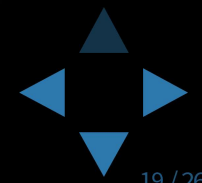
- Counting the number of occurrences of the word "foundme" in a text file:

```
StreamReader streamReader =
 new StreamReader(@"..\..\somefile.txt");
int count = 0;
string text = streamReader.ReadToEnd();

int index = text.IndexOf("foundme", 0);
while (index != -1)
{
   count++;
   index = text.IndexOf("foundme", index + 1);
}

Console.WriteLine(count);
```

What is missing in this code?



Telerik Academy

Counting Word Occurrences

Demo

Telerik Academy

Reading Subtitles – *Example*

- We are given a standard movie subtitles file:

```
.....

{2757}{2803} Allen, Bomb Squad, Special Services...
{2804}{2874} State Police and the FBI!
{2875}{2963} Lieutenant! I want you to go to St. John's Emergency...
{2964}{3037} in case we got any walk-ins from the street.
{3038}{3094} Kramer, get the city engineer!
{3095}{3142} I gotta find out a damage report. It's very important.
{3171}{3219} Who the hell would want to blow up a department store?

.....
```


Telerik Academy

Fixing Subtitles – *Example*

- Read subtitles file and fix it's timing:

```
try
{
  var streamReader = new StreamReader("source.sub");
  var streamWriter = new StreamWriter("fixed.sub");

  string line;
  while ((line = streamReader.ReadLine()) != null)
  {
    streamWriter.WriteLine(FixLine(line));
  }
}
finally
{
  streamReader.Close();
```

FixLine(line) perform fixes on the time offsets: multiplication or/and addition with constant

Telerik Academy

Fixing Movie Subtitles

Demo

Telerik Academy

Summary

- Streams are the main I/O mechanism in .NET
- The `StreamReader` class and `ReadLine()` method are used to read text files
- The `StreamWriter` class and `WriteLine()` method are used to write text files
- Always put file handling in `using(…)` block
- Exceptions are unusual events or error conditions
  - Can be handled by `try-catch-finally` blocks

Telerik Academy

Text Files

Questions?