

Demonstrates opening/creating a file for writing and truncating its length to 0 bytes.



```
/*
C# Programming Tips & Techniques
by Charles Wright, Kris Jamsa

Publisher: Osborne/McGraw-Hill (December 28, 2001)
ISBN: 0072193794
*/
// StrmWrit.cs – Demonstrates opening/creating a file for
writing and truncating
// its length to 0 bytes.
// Compile this program with the following command line:
// C:>csc StrmWrit.cs
using System;
using System.IO;

namespace nsStreams
{
public class StrmWrit
{
static public void Main ()
{
FileStream strm;
// Open or create the file for writing
try
{
strm = new FileStream (“./write.txt”, FileMode.OpenOrCreate,
FileAccess.Write);
```

```

}
// If the open fails, the constructor will throw an exception.
catch (Exception e)
{
    Console.WriteLine (e.Message);
    Console.WriteLine ("Cannot open write.txt for writing");
    return;
}
// Truncate the file using the SetLength() method.
strm.SetLength (0);
Console.WriteLine ("Enter text. Type a blank line to exit
");
// Accept text from the keyboard and write it to the file.
while (true)
{
    string str = Console.ReadLine ();
    if (str.Length == 0)
        break;
    byte [] b; // = new byte [str.Length];
    StringToByte (str, out b);
    strm.Write (b, 0, b.Length);
}
Console.WriteLine ("Text written to write.txt");
// Close the stream
strm.Close ();
}
//
// Convert a string to a byte array, adding a carriage
return/line feed to it
static protected void StringToByte (string str, out byte [] b)
{
    b = new byte [str.Length + 2];
    int x;
    for (x = 0; x < str.Length; ++x) { b[x] = (byte) str[x]; } //
Add a carriage return/line feed b[x] = 13; b[x + 1] = 10; } }
} [/csharp]

```