# Streams and Input

Up until now, all of information that has used in our programs has been entered by the user at run time.
In this section we will read info from files that have been created before the program run.

Reading in a file is a common and extremely useful task for a program.
Imagine being able to read in a huge file of statistics that has been download from the internet.
Much more convenient that have the user type in the data themselves.

VB.Net programs read (and write) data to and from files using using connections called Streams.

input stream
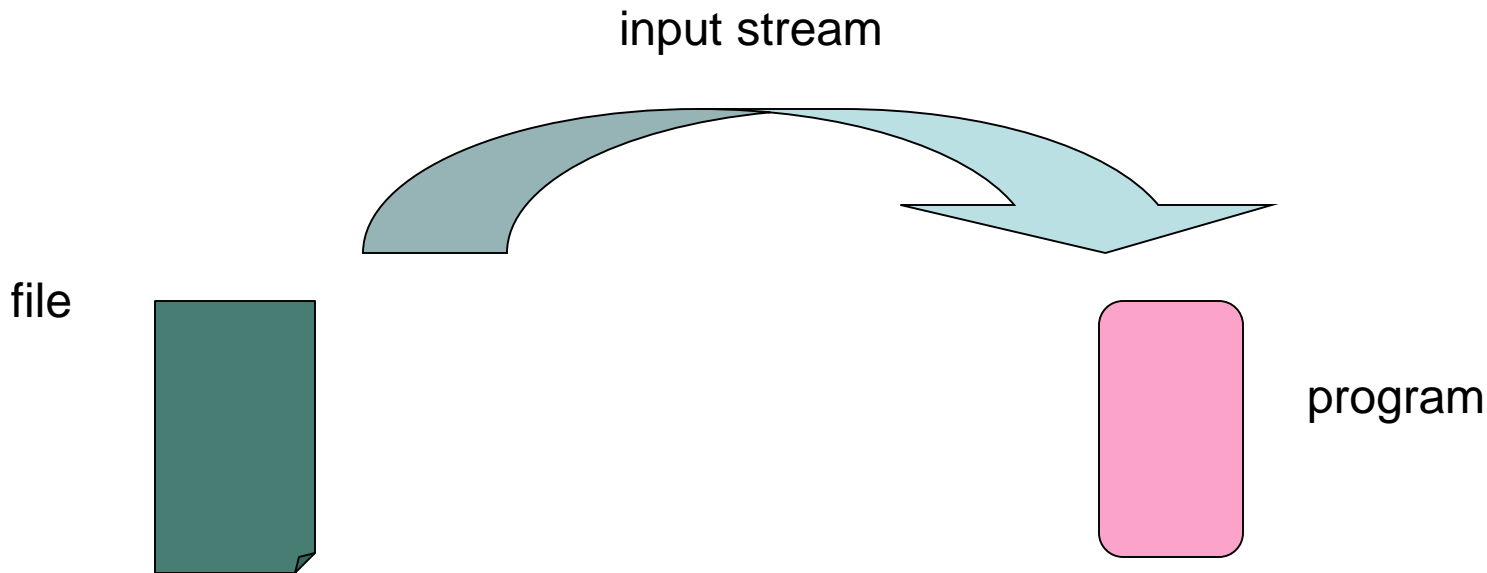
file

program

Input From a File

To make a connection to a file, a program must first know where to find the program.
To do this we will create a String variable which we will call path that lists the path (go figure) to where the file is stored.
This path can be on the hard drive, on a floppy or cd or even on the web.

```
Dim path As String = "C:\Documents and Settings\James\My Documents\Visual Studio 2005\Projects\FirstInput\FirstInput\data.txt"
```
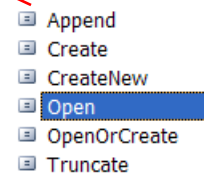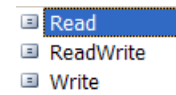
To make a connection between the file and the program we will use something called the FileStream class to make the link. In order to do this we must first import System.IO. This import statement goes at the very beginning of the program, even before the form declaration.

```
Imports System.IO
Public Class Form1
```

Now we will create an object of type FileStream.

Read
ReadWrite
Write

```
Dim fs As New FileStream(path, FileMode.Open, FileAccess.Read)
```

first argument is the file location.

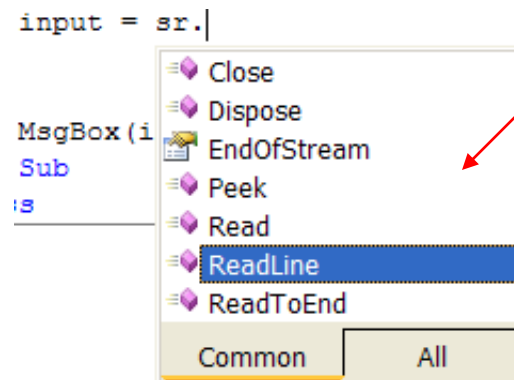| Append |
|--------|
| Create |
| CreateNew |
| Open |
| OpenOrCreate |
| Truncate |

StreamReader

Now that we have a connection between the File and the program we need to create one more object that we create a reader between the FileStream and the program so that we can actually read from the file.

```vb
Dim sr As StreamReader = New StreamReader(fs)
```

sr is an object of type StreamReader which has certain methods that allow for reading.

```vb
Dim input As String
input = sr.ReadLine
```

```vb
input = sr.|
```

```
MsgBox (i
Sub
```

Close	
Dispose	
EndOfStream	
Peek	
Read	
ReadLine	
ReadToEnd	

Common	All

The StreamReader class has other functions that allow it to read in different sorts of ways.

Overview of Connections

```
Dim path As String = "C:\Visual Studio 2005\Projects\FirstInput\data.txt"

Dim fs As New FileStream(path, FileMode.Open, FileAccess.Read)

Dim sr As StreamReader = New StreamReader(fs)


Dim input As String
input = sr.ReadLine
```
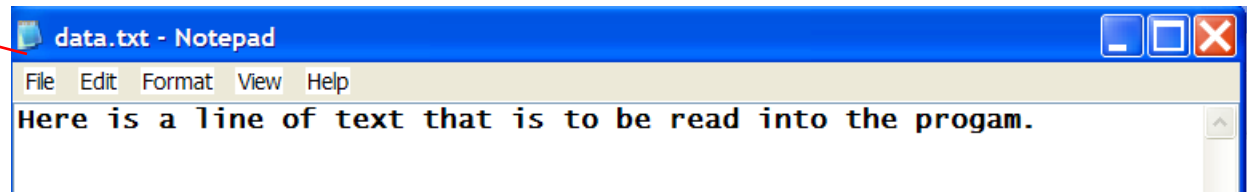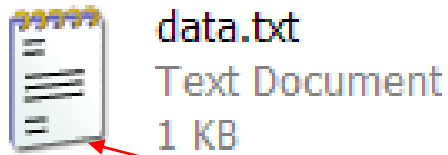
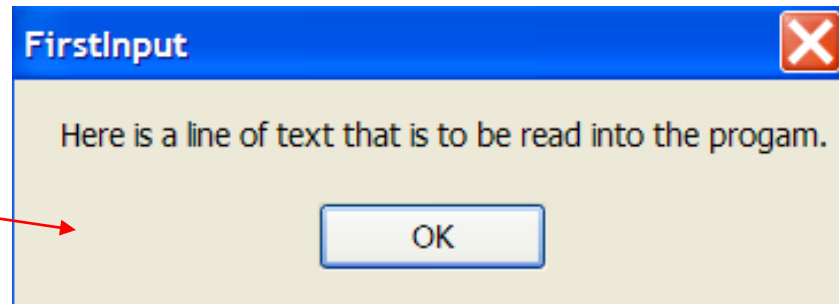FileStream object connects to a file (file path)

StreamReader object connects to a FileStream

StreamReader object has functions that perform reading tasks.

Reading Input From A Text File
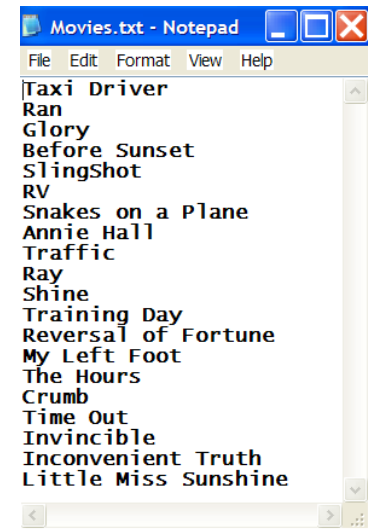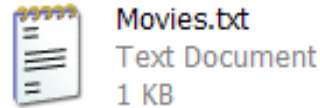
data.txt
Text Document
1 KB

data.txt - Notepad

File Edit Format View Help

Here is a line of text that is to be read into the progam.

MsgBox(input)

FirstInput

Here is a line of text that is to be read into the progam.

OK

Read Whole Document

Movies.txt
Text Document
1 KB

```
Taxi Driver
Ran
Glory
Before Sunset
SlingShot
RV
Snakes on a Plane
Annie Hall
Traffic
Ray
Shine
Training Day
Reversal of Fortune
My Left Foot
The Hours
Crumb
Time Out
Invincible
Inconvenient Truth
Little Miss Sunshine
```
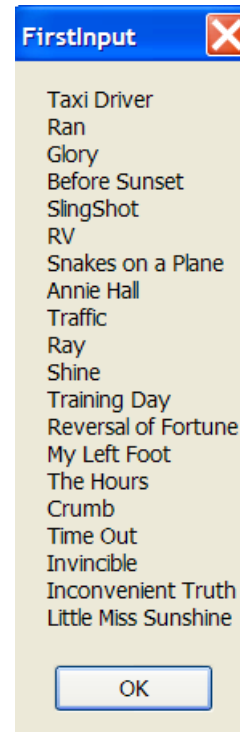
If we switched the document being read into the program by changing the path so it reflected the new file name, the output would look like this…

FirstInput

Taxi Driver

OK
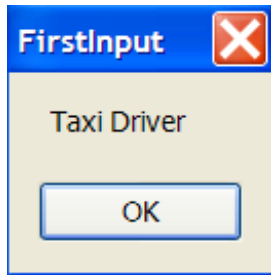
If we made one little change, however to the StreamReader

function so that it read sr.ReadToEnd the output would read as follows:

FirstInput

Taxi Driver
Ran
Glory
Before Sunset
SlingShot
RV
Snakes on a Plane
Annie Hall
Traffic
Ray
Shine
Training Day
Reversal of Fortune
My Left Foot
The Hours
Crumb
Time Out
Invincible
Inconvenient Truth
Little Miss Sunshine

OK

```
Dim input As String
input = sr.ReadToEnd
MsgBox(input)
```

EndOfStream

Within the StreamReader class there is a function which reads a single character at a time and then moves to the next position in the string.

There is also another function which tells the program whether the program has reached the end of data contained within a document. This is a boolean function called EndOfStream.

We can use this in a while loop with the not operator in it. The following syntax reads as:
Keep doing whatever is in the loop until you reach the end.

```
While Not (sr.EndOfStream)
    input = input & sr.Read & wrap
    count += 1
End While


MsgBox("There are " & count & " characters in the document")
```


CountCharacters

There are 236 characters in the document

OK
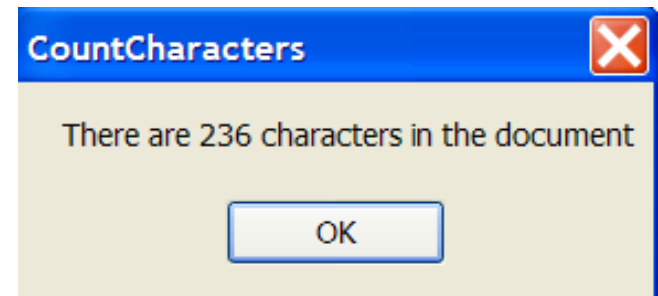
HomeWork

Create a text file called numbers.dat with many integers written one to a line.
Import the file into a program called NumberReader.
Tally the numbers and divide by the number of numbers to find an average.
Output the average in a message box. Include a statement similar to the following.
e.g.. The average of these 25 numbers is 92.4 .

The Code…try to resist peeking until after you have tried it on your own.

```vbnet
Imports System.IO

Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As Syste
        Dim path As String = "C:\Documents and Settings\James\My Documents

        Dim fs As New FileStream(path, FileMode.Open, FileAccess.Read)
        Dim sr As StreamReader = New StreamReader(fs)
        Dim tally As Integer = 0
        Dim average As Decimal
        Dim count As Integer = 0

        While Not (sr.EndOfStream)
            tally = tally + Val(sr.ReadLine)
            count += 1
        End While
        average = tally / count

        MsgBox("The average of these " & count & " numbers is " & average)

    End Sub
End Class
```

CountCharacters

The average of these 37 numbers is 59.1081081081081

OK