UNIVERSITY OF Hull

File Handling

08101 Programming 1
C# Programming

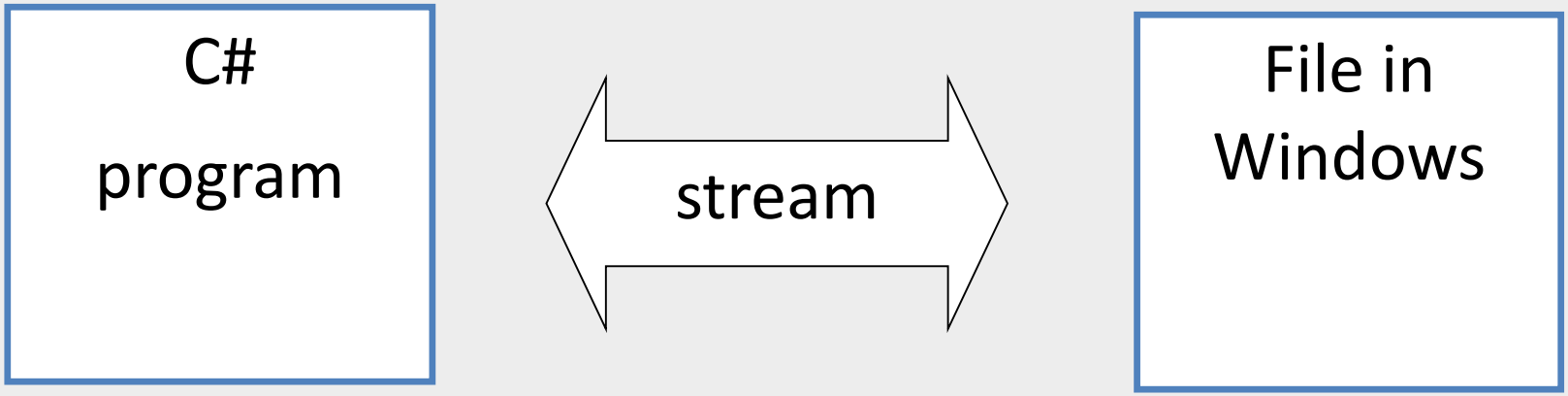
Files

- At the moment when our program stops all the data in it is destroyed
- We need a way of persisting data from our programs
- The way to do this is to use files
- The host computer provides the file storage and the C# System library provides program constructions to use this

Files and Operating Systems

- A file is actually managed by the operating system
- The program uses the ability of the operating system to perform the input/output
- It must do this in a way which is independent of the operating system itself

Files and Streams



- The stream provides the link between a program and a file
- The program sets the value of a stream variable to represent a link to a file

Creating a Stream

```
StreamWriter writer ;
writer = new StreamWriter("test.txt");
```

- The first line creates a stream reference variable called `writer`
- The second line creates a stream and makes `writer` refer to it
- The file being opened is called `test.txt`
- It is being opened for writing

Writing to a Stream

```
writer.WriteLine("hello world");
```

- The `StreamWriter` provides methods that can be used to make it do things
- We have seen this before: the `Console` provides a `WriteLine` method
- The line above would add the line "`hello world`" to the file `test.txt`
- Further calls will add successive lines

Closing a Stream

```
writer.Close();
```

- The `Close` method is called to close the stream and save the file
- It is very important that streams are closed when you have finished with them:
  - An output buffer may need to be emptied
  - If your program has the file other programs may not be able to use it

Using the Stream Classes

- The Stream classes are part of the C# system library, but they are part of a different *namespace*
- A namespace is a part of the system where particular names have particular meaning
- You can regard them as a directory of services, where you can find things you want to use

System.IO Namespace

- The `StreamWriter` class is described in the `System.IO` namespace
- We have to tell the compiler to look in this namespace when we want to use the `StreamWriter` class
- The command to tell the compiler to look in a particular namespace is `using`

Importing the System.IO Namespace

```
using System.IO;
```

- This command tells the compiler to look in the `System.IO` namespace to find the name of classes that are used in the program
- The command is given right at the top of your program file

Using Fully Qualified Names

```
System.IO.StreamWriter writer;
```

- If you don't import the namespace you can still access library objects by using the *Fully Qualified* name of the resource
- This involves putting the namespace in front of the name of the type you want

Namespace Nesting

- Namespaces can be nested
- This allows a tree structure of namespaces to be created
- You have already seen this, in that the IO namespace is nested inside the System namespace
- For a large project you can design your own set of namespaces

File and Directories/Folders

- An operating system does not store all the files in the same place
  - This would be very hard to use
- Instead files are stored in *directories* or *folders*
- A directory is a special kind of file that holds information about files on the system

Directory Trees

- A Directory can hold a reference to another directory
- This means that you can build up a "directory tree" to structure the storage of your files
- In Windows the "root" of the directory tree is always a particular drive
- Windows drives are identified by letter

Understanding a Path

```
c:\data\2007\november\sales.txt
```

- This shows how the path to a file in a directory can be expressed:
  - Start at drive **`C`**
  - Look in directory **`data`**
  - Look in the directory **`2007`** inside that
  - Look in the directory **`November`** inside that
  - Find the file called **`sales.txt`**

File Paths in C# Programs

- When you open a file you give the *path* to the file that you want to use
- This is given as part of the filename that you use when you create the stream
- If you just give the filename the program will use the directory where the program is running
- This is a very bad place to put files…

A Path as a string

```
string path;
path = @"c:\data\2007\november\sales.txt";
```

- You can use a string variable to hold the path to a file
- Because the path separator (the backslash character) is the escape character you need to format the string carefully

File reading

```
TextReader reader =
                    new StreamReader("Test.txt");
string line = reader.ReadLine();
Console.WriteLine (line);
reader.Close();
```

- To read from a file you use an input stream
- If the file does not exist an exception is thrown

Reading an Entire File

```
StreamReader reader;
reader = new StreamReader("Test.txt");
while (reader.EndOfStream == false)
{
    string line = reader.ReadLine();
    Console.WriteLine(line);
}
reader.Close();
```

- The `EndOfStream` property lets a program detect when the end of the file has been reached

Summary

- Streams are used to connect programs to files
- You can create streams for reading and writing
- They provide methods which perform the input and output
- A file path gives the location of a file in a particular folder on a particular disk