

# 19

## Files and Streams

© 2006 Pearson Education, Inc. All rights reserved.

### OBJECTIVES

In this chapter you will learn:

- To create, read, write and update files.
- The C# streams class hierarchy.
- To use classes `File` and `Directory` to obtain information about files and directories on your computer.
- To become familiar with sequential-access file processing.
- To use classes `FileStream`, `StreamReader` and `StreamWriter` to read text from and write text to files.
- To use classes `FileStream` and `BinaryFormatter` to read objects from and write objects to files.

© 2006 Pearson Education, Inc. All rights reserved.

### 19.1 Introduction

- **Persistent Data**
  - Data maintained in files
- **Create, upload and process data files in C#**
- **Overview of data hierarchy**
- **Overview of FCL's file-processing classes**

© 2006 Pearson Education, Inc. All rights reserved.

### 19.2 Data Hierarchy

- **Data Hierarchy**
  - All data in computers are combinations of 0s and 1s
  - The smallest data item is called a bit (Contains 0 or 1)
  - Bytes are composed of eight bits
    - C# uses the Unicode® character set in which characters are composed of 2 bytes
  - Characters are composed of bits
  - A field is a group of characters
  - A record/file is composed of several related fields
    - A collection of programs designed to create and manage databases is called a database management system (DBMS)

© 2006 Pearson Education, Inc. All rights reserved.



Fig. 18.1 | Data hierarchy.

© 2006 Pearson Education, Inc. All rights reserved.

### 19.3 Files and Streams

- **Files and Streams**
  - C# views each file as a sequential stream of bytes
    - Each file ends either with:
      - An end-of-file marker
      - Specific byte number
        - Recorded in a system-maintained administrative data structure
  - An object is created and a stream is associated with the object when a file is opened
  - The runtime environment creates three stream objects:
    - `Console.In` (Refers to the standard input stream object)
      - Enables a program to input data from the keyboard
    - `Console.Out` (Refers to the standard output stream object)
      - Enables a program to output data to the screen
    - `Console.Error` (Refers to the standard error stream object)
      - Enables a program to output error messages to the screen

© 2006 Pearson Education, Inc. All rights reserved.

7

C#'s view of an  $n$ -byte file.

© 2006 Pearson Education, Inc. All rights reserved.

8

### 19.3 Files and Streams (Cont.)

- There are many file-processing classes in the FCL
  - Class `StreamReader`
    - For text input from a file
    - Inherits from abstract class `TextReader`
  - Class `StreamWriter`
    - For text output to a file
    - Inherits from abstract class `TextWriter`
  - Class `FileStream`
    - For both input from and output to a file
    - Can be used to write data to and read data from files
    - Inherits from abstract class `Stream`
  - Class `MemoryStream`
    - Enables the transfer of data directly to and from memory
  - Class `BufferedStream`
    - Uses buffering to transfer data to or from a stream
  - Properties of `Console.In` and `Console.Out` are of type `TextReader` and `TextWriter`, respectively

© 2006 Pearson Education, Inc. All rights reserved.

9

### Common Programming Error 19.1

Failure to open a file before attempting to reference it in a program is a logic error.

© 2006 Pearson Education, Inc. All rights reserved.

10

### Performance Tip 19.1

Close each file explicitly when the program no longer needs to reference the file. This can reduce resource usage in programs that continue executing long after they finish using a specific file. The practice of explicitly closing files also improves program clarity.

© 2006 Pearson Education, Inc. All rights reserved.

11

### Performance Tip 19.2

Releasing resources explicitly when they are no longer needed makes them immediately available for reuse by other programs, thus improving resource utilization.

© 2006 Pearson Education, Inc. All rights reserved.