

Copyright © 2015 by The Learning House.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of The Learning House. For permission requests, write to The Learning House, addressed “Attention: Permissions Coordinator,” at the address below.

The Learning House
427 S 4th Street #300
Louisville KY 40202

System.IO

.NET Cohort

Coding Bootcamp

Lesson Goals

- Explore the System.IO namespace
- Learn how to manipulate files and directories using C#

Key Members of System.IO

Class Type	Usage
BinaryReader BinaryWriter	Allow you to store and retrieve primitive data types (int, bool, string, etc) as binary values
BufferedStream	Temporary storage for a stream of bytes that you can store physically (on the disk, database, etc) at a later time
Directory DirectoryInfo	Allows manipulation of the machines directory structure
DriveInfo	Provides information about the drives on a machine
File FileInfo	Manipulate a machine's set of files
FileStream	Random file access (ex: search/seek) with data as a stream of bytes
FileSystemWatcher	Monitor directories and files for changes
Path	Perform operations on strings that contain file or directory path information (platform neutral)

Key Members of System.IO (2)

Class Type	Usage
StreamWriter StreamReader	Store and retrieve textual information to or from a file. You must read the file sequentially
StringWriter StringReader	Similar to StreamWriter/StreamReader, however the storage is a string buffer rather than a physical file

DirectoryInfo and FileInfo Properties (Both have these)

Property	Usage
Attributes	Gets or sets attributes associated with the current file or directory (ex: read-only, encrypted, hidden, compressed)
CreationTime	Get or set the create time of the file or directory
Exists	Returns true/false based on whether the file/directory exists
Extension	Retrieves only the string of the extension (ex: .csv, .doc)
FullName	Gets the full path of the directory or file
LastAccessTime	Gets or sets the time the current file/directory was last accessed
LastWriteTime	Get/set the time when it was last written to
Name	Returns just the name of the file or directory (not the full path)

DirectoryInfo Type

- The `Directory`(static) and `DirectoryInfo` (use *new()*) classes contain a set of members used for creating, moving, deleting, and enumerating (looping) over directories and subdirectories

DirectoryInfo Members

Member	Usage
Create() CreateSubdirectory()	Creates a directory (or set of subdirectories) for a given path
Delete()	Deletes a directory and all of its contents
GetDirectories()	Returns an array of DirectoryInfo objects that represent the subdirectories in a directory
GetFiles()	Returns an array of FileInfo objects that represent the files in a directory
MoveTo()	Moves a directory and its contents to a new path
Parent()	Retrieves the parent of the directory
Root	Gets the root portion of the path (ex: C:\)

DirectoryApp

DEMO

DriveInfoApp

DEMO

FileInfo Members

Member	Usage
AppendText()	Creates a StreamWriter that appends text to a file
CopyTo()	Copies an existing file to a new file
Create()	Creates a new file and returns a FileStream object to work with the new file
CreateText()	Creates a StreamWriter that writes to a new text file
Delete()	Deletes the file
Directory	Get the parent directory
DirectoryName	Get the full path of the parent directory
Length	Get the size of the current file
MoveTo()	Move the file to a new location, optionally with a new file name

FileInfo Members (2)

Member	Usage
Name	Gets the name of the file
Open()	Opens a file with optional settings
OpenRead()	Opens a read only FileStream
OpenText()	Creates a StreamReader object that reads from an existing text file
OpenWrite()	Create a write-only FileStream

Using Statements

- Using will automatically dispose and close of some framework objects.
- When working with things like files and database we can “lock” resources, which means no one else can use them until we release them
- Using statements allow us to automatically release resources when we are done

File Create With/Without Using

Without Using

```
FileInfo f = new FileInfo(".Test.dat");  
FileStream fs = f.Create();  
  
// write data to file using the file stream  
  
fs.Close();
```

With Using

```
FileInfo f = new FileInfo(".Test.dat");  
using (FileStream fs = f.Create())  
{  
    // write data to file using the file stream  
} // automatic close
```

Simple File IO, StreamWriterReaderApp

DEMO

FileMode Options

Member	Usage
CreateNew	Tells the OS to make a new file, if it exists already, error is thrown
Create	Tells the OS to make a new file, if it exists already, file is overwritten
Open	Open the file, if it doesn't exist, an error is thrown
OpenOrCreate	Opens if the file exists, create if it doesn't
Truncate	Opens an existing file and clears its content
Append	Open a file and move to the end of the file to start writing there. If the file doesn't exist a new one is created

FileAccess Options

Member	Usage
Read	Read Only
Write	Write Only
ReadWrite	Read or Write

FileShare Options

Member	Usage
Delete	Allows others to delete the file while open
None	Don't allow anyone else to read the file while you are working on it
Read	Allows others to reading the file while open
Write	Allow writing the file while open
ReadWrite	Allows others to read or write to the file while open