# Working with CSV Files

# Project Set Up

## Setup Steps for Visual Studio

From within VS (using point and click):

* Open the WorkingWithDataAndFiles solution
* Add console project called csv-demo.csproj to the solution
* Use NuGet to get a reference to the CSVHelper library
* Add a file called movies.csv to the project folder
* Select the movies.csv file in the Solution Explorer window.
* Locate the “Copy To Output Directory” property in the Properties window and select the “Copy Always” option.

## Setup Steps for Visual Studio Code

From Command Prompt:

* Add a new console project to the demos solution by typing:

dotnet new console --output csv-demo

dotnet sln add csv-demo

* Include the CSVHelper package in the csv-demo project by typing:

dotnet add package CsvHelper

* Add a file called movies.csv to the project folder
* Open the csv-demo folder in Visual Studio Code
* Use Visual Studio Code to open the csv-demo.csproj file.
* Open the project file and add an <ItemGroup> element just before the closing </Project> tag
* Add the following to the newly added <ItemGroup> tag:

<None Update="movies.csv">

<CopyToOutputDirectory>Always</CopyToOutputDirectory>

</None>

* Save your changes

## Now do the following regardless of project type:

* Add the following content to the movie.csv file:

Title,ReleaseYear

Star Wars IV,1977

Avatar,2009

Inception,2010

Interstellar,2014

Men in Black,1997

* Replace the Program.cs file’s content with the following:

using System.Globalization;

namespace csvFile\_demo

{

internal class Program

{

static void Main(string[] args)

{

// Reading and displaying a list of people from a CSV file

using (var sr = new StreamReader("movies.csv"))

using (var reader = new CsvReader(sr, CultureInfo.InvariantCulture))

{

var list = reader.GetRecords<Movie>().ToList();

//list.ForEach(m => Console.WriteLine($"{m.Title} is {m.ReleaseYear}"));

foreach (Movie m in list)

{

Console.WriteLine($"{m.Title} is {m.ReleaseYear}");

}

}

// Writing a list to a CSV file

var more\_movies = new Movie[] {

new Movie { Title = "2001: A Space Odyssey", ReleaseYear = 1968 },

new Movie { Title = "Dark Star", ReleaseYear = 1975 },

new Movie { Title = "The Martian", ReleaseYear = 2015 }

};

using (var sw = new StreamWriter("updated\_movies.csv"))

using (var writer = new CsvWriter(sw, CultureInfo.InvariantCulture))

{

writer.WriteRecords(more\_movies);

}

// Reading records into dynamic class (NB: every property value will be a string!)

Console.WriteLine();

using (var sr = new StreamReader("movies.csv"))

using (var reader = new CsvReader(sr, CultureInfo.InvariantCulture))

{

var list = reader.GetRecords<dynamic>().ToList();

foreach (var m in list)

{

Console.WriteLine($"{m.Title} is {m.ReleaseYear})");

}

}

}

}

}

* Review the code and try to work out what it’s doing.
* Build and run the code and confirm it behaves as you expected.