using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace MoveTorrents

{

class Program

{

private static string Default\_Torrent\_Directory = @"C:\Users\" + Environment.UserName + @"\Downloads\";

private static string Domain\_User\_Directory = @"C:\Users\" + Environment.UserName + "." + Environment.UserDomainName + @"\Downloads\";

private static string Default\_Destination\_Directory = @"\\MyWinServer\Torrents\";

static void Main(string[] args)

{

// Important information

Console.WriteLine("This program will help you watch a folder for certain file extensions.");

Console.WriteLine("\nOnce a file is found with the given extension it will be moved to a destination folder.");

Console.WriteLine("\nBy default the watched folder is the logged in user's Download directory.\n\n");

// Variables

string Watched\_Folder = "";

string Destination\_Directory = "";

string Extension\_To\_Watch = ".torrent";

char Key\_Pressed = ' ';

#region Get Folder to Watch

// Check if the user's folder has .DOMAINNAME (i.e. a local account already exists with same name as Domain User)

if (Directory.Exists(Domain\_User\_Directory))

Watched\_Folder = Domain\_User\_Directory;

// If not check just the username

else if (string.IsNullOrEmpty(Watched\_Folder) && Directory.Exists(Default\_Torrent\_Directory))

Watched\_Folder = Default\_Torrent\_Directory;

// If a folder was found

if (!string.IsNullOrEmpty(Watched\_Folder))

{

do

{

// Confirm with the user that this is the folder they want to watch

Console.WriteLine("Download Directory found at: " + Watched\_Folder);

Console.WriteLine("Is this the folder you want to watch? (Y)es (N)o");

Key\_Pressed = Console.ReadKey().KeyChar;

// Validate User's Input

if (!Is\_Yes\_Or\_No(Key\_Pressed))

Console.WriteLine("\nIncorrect input. You must enter 'y' or 'n'.");

// Keep asking until the user enters valid input

} while (!Is\_Yes\_Or\_No(Key\_Pressed));

// If its not then erase it

if (Key\_Pressed == 'N' || Key\_Pressed == 'n')

Watched\_Folder = "";

}

// If no watched folder is picked then have the user input the folder to watch

if (string.IsNullOrEmpty(Watched\_Folder))

{

Console.WriteLine("\nDownload Directory Not found! You will have to enter it manually.");

Console.WriteLine("\nBelow are all the user's found on this pc.\n");

// Display all user folders to help

var User\_Folders = Directory.GetDirectories(@"C:\Users\");

Console.WriteLine(@"Users in C:\Users\ :");

foreach (string folder in User\_Folders)

Console.WriteLine(folder);

// While the user inputs incorrent paths

while (!Directory.Exists(Watched\_Folder))

{

// If the user entered a path let them know it was wrong

if (!string.IsNullOrEmpty(Watched\_Folder))

Console.WriteLine("\nThe Path you entered doesn't exist!");

// Get the path from the user

Console.WriteLine("\nPlease enter the path to the watched folder: ");

Watched\_Folder = Console.ReadLine();

}

}

#endregion

#region Get the File Extension to watch

do

{

// Get the file extension to move

Console.WriteLine("\n\nWhat type of file do you want to move?");

Console.WriteLine("\nIs the default file extension to move '.torrent' ok? (Y)es (N)o");

Key\_Pressed = Console.ReadKey().KeyChar;

// While the user inputs incorrent input

} while (!Is\_Yes\_Or\_No(Key\_Pressed));

// If its not then ask the user to input the correct one

if (Key\_Pressed == 'N' || Key\_Pressed == 'n')

{

bool Correct = false;

do

{

Console.WriteLine("\n\nWhat file extension do you want to move? (Include the . i.e. '.jpg')");

Extension\_To\_Watch = Console.ReadLine();

do

{

Console.WriteLine("\n Confirm you want to watch files with this extension: " + Extension\_To\_Watch + " (Y)es or (N)o");

Key\_Pressed = Console.ReadKey().KeyChar;

// If it is set Correct to true and exit the loop

if (Key\_Pressed == 'Y' || Key\_Pressed == 'y')

Correct = true;

// While the user inputs incorrect input

} while (!Is\_Yes\_Or\_No(Key\_Pressed));

// While the user says "no this is not the file extension I want"

} while (!Correct);

}

#endregion

#region Get the Destination Folder

// Confirm destination location

Console.WriteLine("\n\nIs the destination folder located at : " + Default\_Destination\_Directory + " ? (Y)es or (N)o");

Key\_Pressed = Console.ReadKey().KeyChar;

// If destination is not in the default location

if (Key\_Pressed == 'N' || Key\_Pressed == 'n')

{

// While the user inputs incorrent paths

while (!Directory.Exists(Destination\_Directory))

{

// If the user entered a path let them know it was wrong

if (!string.IsNullOrEmpty(Destination\_Directory))

Console.WriteLine("\nThe Path you entered doesn't exist!");

// Get the path from the user

Console.WriteLine("\nPlease enter the destination path: ");

Destination\_Directory = Console.ReadLine();

}

}

else

Destination\_Directory = Default\_Destination\_Directory;

// Inform user of settings

Console.WriteLine("\nStarting Up...\n");

Console.WriteLine("Watching Folder: " + Watched\_Folder);

Console.WriteLine("For File Extensions: " + Extension\_To\_Watch);

Console.WriteLine("With The Destination Folder: " + Destination\_Directory);

#endregion

#region Run Forever Moving Files With Given Extension From Watched Folder to Destination Folder

// Run forever

while (true)

{

Console.WriteLine("Searching... [ " + Watched\_Folder + " ]");

// Get all files with extension of .torrent

var Files = Directory.GetFiles(Watched\_Folder, "\*" + Extension\_To\_Watch, SearchOption.TopDirectoryOnly);

// Foreach file move it

foreach (string file in Files)

{

try

{

File.Move(file, Destination\_Directory + Path.GetFileName(file));

Console.WriteLine("Moved File [ " + file + " ]");

Console.WriteLine("To: [ " + Destination\_Directory + Path.GetFileName(file) + " ]");

}

catch (Exception e)

{

Console.WriteLine("Failed to move [ " + file + " ] to [ " + Destination\_Directory + Path.GetFileName(file) + " ]");

Console.WriteLine("Because : " + e.Message);

}

}

// Take it easy on the cpu

Thread.Sleep(3000);

}

#endregion

}

public static bool Is\_Yes\_Or\_No(char c)

{

if (c == 'Y' || c == 'y' || c == 'N' || c == 'n')

return true;

return false;

}

}

}