Directory Syncher Version 1.1 -- Made and maintained by Dan Martineau

Directory Syncher is a minimal synchronization/backup solution intended to be paired with some form of scripting such as BASH, Powershell, or any other form of shell scripting. It has been written in Java and has been tested on both Windows and Linux. While it has not been tested on Mac OS, the program is engineered to be fully platform independent and should run anywhere that Java 8 or above will run.

Java must be installed in order to run this program.
See Oracle's website to download Java: https://www.java.com/

The source code for this project is available in the "SourceCode" branch located at:
https://github.com/justdan325/Directory-Syncher/tree/SourceCode

---

For a quick reference guide, launch the program with the --help flag:

```
java -jar synch.jar --help
```

--Permissions--

The heart of the program is the permissions, which are what allow the program to add, modify, and delete files. The permissions are represented as three bits:

```
000
```

Here the first bit is "read," the second is "modify," and the third is "delete." They are either set to "0" for off or "1" for on.

```
100 --> [ <read on>  <modify off> <delete off> ] (this is the default)
011 --> [ <read off>  <modify on> <delete on> ]
101 --> [ <read on>  <modify off> <delete on> ]
etc.
```

-Read allows files that are not in one directory to be copied to the other directory.
-Modify allows files that have been updated (such as a spreadsheet) to have the newest version copied over to replace an older version.
-Delete allows files that do not exist in one directory to be deleted from the other directory.

--Flags--

Flags that can be included in a synch job are as follows:

-u : A unidirectional run. This means that the program will only run in one direction insted of both directions. This is useful for backup drives where the end goal is not to synchronize, but to have a clone of a directory and its sub directories.
-l : This will save a log file in the "synchlogs" sub directory.
-v : This will make the program run in verbose mode, which allows one to see the program's progress in real time. It is recommended to use this flag for large directories with many files.

--Operating Procedures--

e.g. Use "C:\Users\Joe\Desktop\BackThisUp\" as opposed to "Desktop\BackThisUp\" on Windows. On Linux, use "/home/joe/Desktop/BackThisUp/" as opposed to "~/Desktop/BackThisUp/"

When running a normal, bidirectional synch job, the program runs twice. It will first run from the primary directory to the secondary directory. Once finished, it will then run from the secondary directory to the primary directory unless the "-u" flag has been included. Keep this in mind if you plan to use this program with critical data! If you have delete turned on, files in the secondary that are not in the primary WILL BE DELETED upon the first run.

The default procedure is as follows:

```
java -jar synch.jar [primary directory] [secondary directory]
```

Here the primary directory is compared with the secondary directory, and any files and sub directories that exist within the primary that do not exist within the secondary will be copied over to the secondary. Remember that this is due to the default permissions being "100." See the permissions section for clarification.

e.g. java -jar synch.jar C:\Users\Joe\Desktop\BackThisUp\ E:\JoeBackup\

To run with custom permissions, include the permission bits before the directory paths. Be very careful! If you have delete turned on, files in the secondary that are not in the primary WILL BE DELETED even though the drives are being synchronized.

```
java -jar synch.jar [permissions] [primary directory] [secondary directory]
```

e.g. java -jar synch.jar 111 C:\Users\Joe\Desktop\BackThisUp\ E:\JoeBackup\

To run with flags and default permissions, inculde the flags before the directory paths.

```
java -jar synch.jar -[flags] [primary directory] [secondary directory]
```

e.g. java -jar synch.jar -vlu C:\Users\Joe\Desktop\BackThisUp\ E:\JoeBackup\

To run with both flags and custom permissions, format as follows:

```
java -jar synch.jar -[flags] [permissions] [primary directory] [secondary directory]
```

e.g. java -jar synch.jar -v 111 C:\Users\Joe\Desktop\BackThisUp\ E:\JoeBackup\

--The "synchrc" File--

The synchrc files are automatically generated upon a run of the program. These files are plain text and can be edited to alter the permissions for specific files and sub directories. For instance if you are running a synch job with read, modify, and delete all set, but you wish to have a particular file (E:\JoeBackup\FolderOfGoodies\fileToBeIgnored.jpg) ignored, simply add the following line to synchrc:

```
000 JoeBackup,FolderOfGoodies,fileToBeIgnored.jpg
```

The first three characters are the custom permissions for that file. "JoeBackup" is the directory that the job is being run on. The comma is the delimiter character for the program (this is in an effort to maintain platform independence). "FolderOfGoodies" is the sub directory the file is in. Lastly, the name of the file appears.

Directories can be treated the same way, but keep in mind that setting custom permissions for a directory will only affect it, NOT its files and sub directories. i.e. 000 will ignore a directory and prevent it and its contents from being copied, but 100, 110, 111, etc, will not affect any of the downstream files.

By default, the synchrc file will have a line included to ignore itself so that the synchrc in the primary directory will not overwrite the synchrc in the secondary directory. It will include two additional lines that prevent "$RECYCLE.BIN" and "System Volume Information" directories from being read on Windows drives. These usually contain meta-data files that should not be read by anything other than Windows and can cause run-time exceptions in the Java Virtual Machine if they are read.

You can include a temporary synchrc file that differs from the main one if you have a very elaborate synchrc file and want to perform a unique run. Simply include either or both of these flags:

```
--rcPrim [custom synchrc file for primary directory]

--rcSec [custom synchrc file for secondary directory]
```

e.g. java -jar synch.jar 111 C:\Users\Joe\Desktop\BackThisUp\ E:\JoeBackup\ --rcPrim C:\Users\Joe\Documents\customrc.txt