

Code is located here: <https://github.com/tursilion/tursicopy>

So the idea, then, is that we have a few modes:

tursicopy src: dest: -- does a backup now from src to dest with default values. No profile is used.

tursicopy /now profile.txt -- does a backup now using the settings in 'profile.txt'

tursicopy /watch -- runs in the background and waits for a removable drive to be installed with a 'tursicopy_profile.txt' - if it finds one, it runs a backup to that drive with the profile.

tursicopy /default -- prints a default profile.txt to the console that you can copy and paste to create a new configuration

Secret mode: **tursicopy /auto profile.txt** -- same as '/now' but enables the UnmountDevice option.

Users should not manually select this mode, you might accidentally unmount your main drive! (And it isn't currently enabled anyway)

Inside a profile.txt file (which can have any name, except in the automatic case):

[Setup]

Setup section has the paths to the destination drive and optional log.

DestPath=C:\Path - sets the root destination path. For an external drive insertion, the drive letter is replaced with the drive inserted (so it MUST be a DOS-style path, ie: C:\target\). For all other modes it's absolute.

LogFile=C:\Path.txt - sets the path to the log, or leave it empty for no log. Again, for an external drive insertion, the drive letter is replaced with the drive inserted. Any old log file is overwritten. **Note:** if you use the EnableDevice feature, the log can not be stored on the device being enabled. This is because the log is opened before the enable takes place (so there is a record if it fails).

The source section allows the selection of multiple source folders. All paths are absolute. There's no way to automatically copy FROM an external drive insertion back to the host.

[Source]

Name1=Path

Name2=Path

... etc...

"Name1" is the name of the folder at the target. Path is the path to copy from. For instance:

MyPics=C:\Pictures

Will copy from C:\Pictures to the backup drive 'X' at X:\MyPics\.* Of course, 'X' will be the contents of 'DestPath'. These paths may be any legal path on the system.

Source folders will be tested for case sensitivity on the first file found. If a folder is case sensitive (for example, a WSL folder), then uppercase letters will be prepended by '^' in the output filename to avoid conflicts with similarly named files that differ only by case. If a file contains a '^', then it will be written as '^'. For example: MyFile.txt -> ^My^File.txt.

You can override this detection by adding a :CS to force case sensitive, or :NOCS to force not case sensitive to the Name. ie: MyPics:NOCS=C:\Pictures

[Filter]

Filter section lets you skip over files by source path – this can be ANY substring.

skip=Q:\oldbackup – specifies a substring to skip. You can repeat this as many times as needed. TEST IT CAREFULLY. If you enable the Verbose flag, you can see which files are skipped.

[Paranoid]

Paranoid section handles mounting and unmounting of drives, as well as a little tuning, for extra paranoia. The app needs administrative rights to mount or unmount, otherwise it does not.

EnableDevice=(string that indicates the device to enable) -- Only used in **/now** mode. When the backup starts, attempt to mount this device and fail if we can not. The string can be obtained from the device manager under Details/Device Instance Path. You will get a string like:

`USBSTOR\DISK&VEN_SEAGATE&PROD_EXPANSION_DESK&REV_0319\2HC015KJ&0`

It must match /exactly/. The app only searches disk devices but nothing else is verified.

UnmountDevice=0 -- if set to 1, then we unmount (actually disable) the device when done if we enabled it using EnableDevice. Note this is not 100%, sometimes Windows decides a reboot is necessary and there's nothing we can do about that. The log will report this case.

FindDrive=(drive name) -- if set, searches for the drive with the specified name (*exact* match except for case-insensitive) and uses that drive for the destination path instead of what is configured. Use this if Windows likes to change your drive letters up on you or you want to ensure the correct drive is attached. (Make sure your drive names are unique!) If Logfile is configured to the same drive as DestPath, then it is also updated, otherwise it is left alone.

PauseOnErrors=1 -- if set to 1, then the program will wait for a keypress on exit if there were any errors, otherwise (or if set to 0) it will just exit.

PauseAlways=0 -- if set to 1, then the program will always wait for a keypress on exit.

Verbose=0 -- if set to 1, then more information is output in the log.

[Tuning]

Optional tuning settings. It is not recommended to change any of these.

Reserve=100 -- megabytes to reserve on the disk - if fewer bytes are free, folders are deleted. 100MB is the default

SaveFolders=5 -- number of old backups to save. This prevents a runaway condition from removing ALL backup folders if the disk fills up. Instead the backup will abort.

TimeSlack=5 -- number of seconds of time difference that's ignored. Less than 2 is a bad idea because that's the default quantum of copied timestamps (ie: they would almost always be different). Set to -1 to have the timestamp ignored (ie: only size will be considered. This means changes that don't change the filesize won't be detected.)

MountDelay=5 -- number of seconds to wait after mounting a drive before proceeding

UnmountDelay=30 -- number of seconds to wait after backup before unmounting a drive

RotateOld=1 -- if 0, don't rotate old backup folders. This would defeat most of the purpose of this tool.

DoBackup=1 -- if 0, don't perform the backup sequence. That would defeat the rest of it.

DeleteOld=1 -- if 0, don't delete orphaned files - that is, files which were removed from the src are not removed from the backup. (Note that 'removed' still means moved to the history folder). There may be cases to argue for this.

isCompressed=0 -- if this is set to 1, then your target drive is using Windows compression. This requires enough space to store the full file plus the compressed version, so the free space calculation will estimate twice the needed space.

NOTE - an issue with network drives is that they are mapped per user (unlike physical drives). So if you run as administrator in order to be able to mount/unmount, then you won't have access to your mapped drives and will get error code 3.

Apparently, you can work around this with linked connections:

```
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System =  
DWORD:EnableLinkedConnections:1
```

Reboot is required. This makes all accounts share mapped drives (which you do not want to do if you have a multi-user system.) Don't do this if you don't need to, though.

NOTE – Case sensitive source drives are now supported, however, the following limitations are in place:

- The sensitivity is detected by using the first file in a source as a test, and checking whether the file, the file all uppercase, the file all lowercase, and the file in mixed case all exist. (ie: for MyFile.txt, it also checks MYFILE.TXT, myfile.txt, and mYfIlE.TxT. If they all exist, as is the case for a normal Windows drive, the source is considered NOT case sensitive. If any one does not, then it IS case sensitive.
- The OUTPUT folder is not tested for case sensitivity. It's assumed it is not, but this probably doesn't matter.
- Case sensitivity is NOT checked for folders, only individual files. If two folders differ only by case, they will still conflict.
- Windows 11 now supports case sensitivity on a per-folder basis. This will only work with Tursicopy if the folder in question is its own unique source.