If you ask robocopy to destroy the destination, then it will destroy the destination

devblogs.microsoft.com/oldnewthing/20161220-00

December 20, 2016

Raymond Chen

A customer reported that Explorer was not showing a folder on their hard drive that they were sure was there.

I asked them to check whether the folder really was there, by going to a command prompt and using the `dir /a` command.

Turns out the folder really was gone.

The customer went back and retraced their steps and reconstructed what happened.

First, the customer created a folder on their `D:` called `D:\backups\fdrive\spreadsheets`.

Next the customer copied two files from their `F:` drive to the `D:\backups\fdrive\spreadsheets` folder.

So far so good.

Next, the customer wanted to copy their entire `F:` drive to the `D:\backups\fdrive` folder, so they performed the following command:

```
robocopy /MIR F: D:\backups\fdrive
```

The customer let this command run for a while, but then the operation started encountering Access denied errors, so they hit `Ctrl`+`C` to stop the robocopy command.

At this point, the customer noticed that the `spreadsheets` folder was gone.

The customer theorized, “I suspect that what happened is that robocopy was matching the directory structure of the `F:` drive against the directory structure of `D:\backups\fdrive`, and since my important spreadsheet files weren’t present in the `F:\spreadsheets` folder on the source, it deleted them from the destination. If I had let the copy run to completion, it presumably would have eventually copied the files from their location on the `F:` drive to the corresponding subdirectory of `D:\backups\fdrive`.”

The customer continued, “From a user perspective, it seems that I *should* have really been alerted by robocopy that the target folder (in this case, `D:\backups\fdrive`) wasn’t empty, and it should have asked for confirmation that I didn’t really want to lose those files (which I didn’t).”

Well, um, yeah, because that’s what the `/MIR` option means.

```
       /E :: copy subdirectories, including Empty ones.
   /PURGE :: delete dest files/dirs that no longer exist in source.
     /MIR :: MIRror a directory tree (equivalent to /E plus /PURGE).
```

The `/MIR` option means that the destination folder should be an exact copy of the source folder. The documentation points out that this is the same as `/E` (copy recursively) combined with `/PURGE` (remove anything from the destination that is not present in the source).

The `/PURGE` behavior is by definition destructive.

If that’s not what you want, then don’t pass the `/MIR` flag.

Raymond Chen

Follow