

# Getting Around Windows 7 "Destination Path Too Long" Error When Deleting Files

Posted At : January 19, 2012 10:15 AM | Posted By : Steve

Related Categories: Personal, Windows

Years ago when I started learning VBA for Excel, I was reading a book by [John Walkenbach](#) and he compared learning macros in Excel to using a remote control - once you learn it you don't know how you lived without it. I would say that is true of scripting in general.

The bad news is that programming is also like using a credit card, inasmuch as you can get yourself into a depth of problem just not possible without it.

I got experience with that first hand recently when one small programming mistake caused a bit of a disaster on my computer.

My plan was that I would write some automated tests to test a vanilla installation of the [Neptune framework](#). As part of this, I back up my CustomTags folder by renaming it "CustomTag\_BAK" - easy enough.

Unfortunately, I made a mistake while making some changes that caused it to copy "CustomTags/\_BAK". Worse yet, I did this using a recursive bit of code (if programming is dangerous, recursion is doubly so). The effect of which was several levels deep of "\_BAK" folders in my CustomTags folder.

Actually, "several" doesn't cut it. It was thousands of levels deep. I'm not sure how many as Windows seemed to stop counting. This seemed an easy problem to solve, however. I just highlighted the folder on clicked delete.

## "Destination Path Too Long"

The problem is that Windows cannot delete a file whose path is longer than 256 characters long. It will let you create files with paths longer than this, but not delete them.

So, how to fix it? Well, I found a [page by Microsoft](#) that suggested some solutions (which I'll simplify a bit).

### Rename the folder names

This is a nice idea, but as I have more than 256 folders, it really won't help.

### Manually move the file to another folder

This is a good solution if the problem is that the folder is just a little bit too long. The problem is that if the folder you move is still too long then you still can't delete it.

So, in my case, I would have to find the deepest folder and start from there. Several minutes of clicking to get there showed me that I was no closer to it then when I started.

### Create a virtual drive

This would work well if my folder depth totaled less than 256 X 26 characters long (that is the character limit times the number of letters in the alphabet and thereby the number of virtual drives I could create). Again, my problem was much worse than this.

At this point, I have to admit that I was feeling a little concerned. Here I start to consider just dumping my computer and restoring from back-up. Unfortunately, due to not noticing the problem quickly enough this will mean that I either restore from a back-up that has the problem I am trying to

fix or from one that is missing some of what I need (or go through a painful process of cherry-picking files).

Nope, there has to be a better way.

Fortunately there is (or this would be a depressing blog entry).

## Move the files with a Script

Someone (I can't find who anymore, sorry) wrote a nice .bat file that (with a little bit of editing) solved by problem. They named the file "recupera.bat", so I kept that name.

recupera.bat:

```
ren c:\ColdFusion9\CustomTags\_BAK\_BAK x

move c:\ColdFusion9\CustomTags\_BAK\_BAK\x c:\ColdFusion9\CustomTags\_BAK

rd c:\ColdFusion9\CustomTags\_BAK\_BAK

ren c:\ColdFusion9\CustomTags\_BAK\x _BAK

repucera.bat
```

So, what does this do?

```
ren c:\ColdFusion9\CustomTags\_BAK\_BAK x
```

This takes "c:\ColdFusion9\CustomTags\\_BAK\\_BAK" and renames it to "c:\ColdFusion9\CustomTags\\_BAK\x". Apparently Windows has no problem renaming paths even if the result is a path that is too long for it to work with.

```
move c:\ColdFusion9\CustomTags\_BAK\_BAK\x c:\ColdFusion9\CustomTags\_BAK
```

This moves all of the files from "c:\ColdFusion9\CustomTags\\_BAK\\_BAK\x" to "c:\ColdFusion9\CustomTags\\_BAK". So, now "c:\ColdFusion9\CustomTags\\_BAK" has two folders: "x" and "\_BAK". The "x" folder still has tons of subfolders, but the "\_BAK" folder is empty.

```
rd c:\ColdFusion9\CustomTags\_BAK\_BAK
```

This just removes the now empty nested "\_BAK" folder.

```
ren c:\ColdFusion9\CustomTags\_BAK\x _BAK
```

This renames the "x" folder (which has tons of nested "\_BAK" folders) back to "\_BAK". Now my situation is the same as when I started except that one folder has been removed.

```
repucera.bat
```

This just executes the .bat again - a nice loop. At some point the "ren" line will throw an exception because the "c:\ColdFusion9\CustomTags\\_BAK\\_BAK" will no longer exist. This will stop the loop. At this point the problem is finally solved.

So, if you ever find yourself in the bind that I created for myself then feel free to take this code and modify it to suit your problem (renaming the path to the location of your problem).

Hopefully I will never need this again.