The Old New Thing

Why does Windows Compressed Folders (Zip folders) reject paths that begin with a slash?

10 Oct 2012 7:00 AM | 20

A customer asked, "Does NTFS support files with a null string as the name?"

No, NTFS does not support files with no name. None of the commonly-used Windows file systems do. Files must have a name. But what a strange question that is. The customer was kind enough to explain why they cared.

"We have a zip file that the Compressed Folders (Zip folders) feature that comes with Windows cannot deal with. When we try to extract the contents of the zip file, we get the error message 'Windows has blocked access to these files to help protect your computer.' We've attached a copy of the file."

The Compressed Folders functionality in Explorer has many known limitations, such as lack of support for ZIP64 and AES encryption. Neither of those applied to the zip file in question, however.

The customer explained what they did. "We created the zip file with a third party zip tool. In particular, after adding a directory tree to the zip file, we renamed the root of the tree to have a blank name. In the zip file we sent you, we added A/file.txt, and then we used the zip tool to rename 'A' to the empty string."

And indeed if you looked at the zip file in a hex editor, the file name was "/file.txt".

Now the pieces fell into place. The Compressed Folders code was blocking the file because it was attempting to perform a directory traversal; specifically, it was trying to drop a file in the root directory. The ZIP Application Note says that the "file name" field consists of "The name of the file, with optional relative path." Note that the path must be relative. The next sentence emphasizes this point: "The path stored should not contain a drive or device letter, or leading slash." Therefore, the zip file is invalid, and the Compressed Folders code is within its rights to reject it. (And one wonders why the zip tool allowed the user to create an invalid zip file.)

It's unclear what the customer was trying to do by renaming "A" to the empty string. So the recommendation back to the customer was "Don't do that."

Blog - Comment List MSDN TechNet

Comments

Rob
10 Oct 2012 7:08 AM
#

Doctor, it hurts when I do this!

Then don't do that!

PkWare
10 Oct 2012 7:12 AM
#

I love the fact that the document was updated little more than a month ago (reformatting). And that there is still pkware around.

Sad story the one of Phil Katz though :(

Joshua
10 Oct 2012 8:05 AM
#

I wonder if it catches symlink directory traversal.

Medinoc
10 Oct 2012 8:38 AM
#

Speaking of ZipFldr.dll, does it now call SHGetInstanceExplorer (blogs.msdn.com/.../8555658.aspx) for its worker threads so a program can refrain from terminating until it's done? Because last time I tried (which was admittedly five years ago), it didn't.

Danny
10 Oct 2012 9:58 AM
#

Get a Word .docx file. Rename it to .zip. Use that tool to rename one file inside the .zip archive to empty name. Rename the .zip back to .docx. Try open it in Word and watch what is happening. (Brought to you by Toy..erm..Danny :P)

The MAZZTer
10 Oct 2012 10:13 AM
#

Joshua: Not sure how that would be a problem. ZIP files can only extract files to the current directory and never a parent one. ZIP files also cannot embed symlinks inside themselves; the format doesn't support it AFAIK. So the only symlink traversal that can be done is with symlinks that already exist on the filesystem. No system-created symlink in Windows does anything like linking to the root dir (mostly just links inside of a user's profile directory), and if the user makes one its their problem.

Josh

10 Oct 2012 10:15 AM
#

Couldn't they have just renamed the root to "."?

Cesar
10 Oct 2012 10:29 AM
#

> It's unclear what the customer was trying to do by renaming "A" to the empty string.

They probably wanted to move the contents one level up. That is, instead of A/file.txt, they wanted file.txt. They probably do not know Unix enough to understand that "/file.txt" means file.txt in the root directory, and probably thought it meant the same as "./file.txt", that is, in the current directory.

foo
10 Oct 2012 10:36 AM
#

Adventures in undefined behaviour, user space

Cesar
10 Oct 2012 10:37 AM
#

@The MAZZTer: "ZIP files also cannot embed symlinks inside themselves; the format doesn't support it AFAIK."

Wrong. Look at the APPNOTE.TXT Raymond linked. I can easily create a zip file with a symbolic link:

```
$ touch file.txt

$ ln -s file.txt link.txt

$ zip --symlinks test.zip link.txt
```

It does recreate the symbolic link when unpacked:

```
$ unzip test.zip

Archive:  test.zip

   linking: link.txt                -> file.txt

finishing deferred symbolic links:

 link.txt               -> file.txt
```

However, I expect the Windows built-in unpacker to ignore these symbolic links.

j b
10 Oct 2012 10:57 AM
#

"Files must have a name. But what a strange question that is."

Files _must_ have a name? Why? I've been working with file systems allowing nameless files. Since a nameless file cannot be identified by name, when the last user closes it, it is deleted. Make as many temp files as you like by opening them specifying an empty name string. Or make sure a named file is deleted as soon as your process has completed its work on the data, by renaming it to an empty string. Makes perfect sense.

Joshua
10 Oct 2012 12:06 PM
#

@j b: I give you the null file name.

docstore.mik.ua/.../ch23_13.htm

Macrosofter
10 Oct 2012 12:14 PM
#

>The Compressed Folders functionality in Explorer has many known limitations, such as lack of support for ZIP64

So this story was before Vista?

k m
10 Oct 2012 12:32 PM
#

@j b: Sure, nameless files are fine. But do they really make sense in the context of an archival program? No.

Bob
10 Oct 2012 1:23 PM
#

@j b: "Files _must_ have a name? Why? I've been working with file systems allowing nameless files. Since a nameless file cannot be identified by name, when the last user closes it, it is deleted."

I prefer the way VMS does it with explicit file attributes when creating a file. This makes it obvious to the developer who comes along next that you are creating a temporary file and it's not a coding bug that you are creating a file with no file name. Or even worse, is the coding error that intermittently generates null file names and has people complaining, "The program ran, but it didn't create a file."

Interestingly, VMS has two different temporary file attributes. When either of these attributes is set, VMS does not create a directory entry for the file. One attribute functions as most people expect; when the last process that has the file open, closes it, the file is deleted by the file system. The second attribute says that when the last process that has the file open, closes it, the file is NOT deleted. IIRC, the file can be deleted by passing the equivalent of the file handle to the file delete system call.

Gabe
10 Oct 2012 10:12 PM
#

j b: Allowing files without a name sounds great up until you try to run your system over a network (diskless workstation, anyone?) -- and then you have to figure out how to get your stateless filesystem to be compatible with all those programs that expect to be able to create temporary files by deleting them as soon as they're created (silly rename?).

Another common problem with nameless files is seeing a large log file and deleting it to free up disk space, but not realizing that the logging process still has it open. The logger will still continue to run, chewing up space until the disk is full, with no way to read the log file or even know that it's growing.

gechurch
10 Oct 2012 10:41 PM
#

@j b

Every sentence Raymond wrote up until the one you quoted clearly sets the context of his statement as being Windows file systems, specifically NTFS. When trying to show everyone how clever you are it's best not to completely miss the point.

Rob
11 Oct 2012 1:21 PM
#

"The Compressed Folders functionality in Explorer has many known limitations, such as lack of support for ZIP64 and AES encryption" and Unicode. How could one forget that glaring omission from a modern OS?

ErikF
11 Oct 2012 3:03 PM
#

@Rob: Every time Microsoft adds additional functionality, typically the EU goes after them for "monopolistic practices". If you got in trouble every time you wanted to add stuff, eventually you'd stop bothering too.

Anyways, there are lots of programs that will do the additional stuff for you if you need to. As Raymond says, "Don't be helpless!"

Alexander Sklar
15 Oct 2012 5:17 AM
#

@Rob: you'll be happy to learn we've added Unicode support for decompressing zip files in Windows 7 (support.microsoft.com/.../2704299) and Windows 8.