

Kerbal Space Program - Bug #13268

Part fails to load when next folder is read-only

11/19/2016 03:29 PM - WildLynx

Status:	Updated	Start date:	11/19/2016
Severity:	Normal	% Done:	10%
Assignee:			
Category:	Application		
Target version:			
Version:	1.2.1	Language:	English (US)
Platform:	Linux	Mod Related:	No
Expansion:			

Description

Situation:

There is a mod (or even stock) folder with "Parts" folder.

There is "SomePartKind" folder in the "Parts" folder.

There are many "PartName" folders in the "SomePartKind" folder.

One of the "PartName" folders is set to ORX-GRX-ORX access mode, eg set to be read-only.

Effect: Parts from **previous** "PartName" FAIL TO LOAD! Not from folder, that is RO, but folder, that is previous to it.

History

#1 - 11/19/2016 03:30 PM - WildLynx

correction: URX-GRX-ORX

#2 - 11/19/2016 03:40 PM - WildLynx

Problem lingers, even with attributes reverted to ARWX

Cleared Module manager cache - restored part.

Module Manager issue or stock issue?

#3 - 11/20/2016 05:55 PM - JPLRepo

- Status changed from New to Needs Clarification

Is this a modded install?

Can you please reproduce the problem in a stock install without any mods and please provide a log file.

Details of how to do this is within the wiki page.

<http://bugs.kerbalspaceprogram.com/projects/ksp/wiki>

Thanks

#4 - 12/11/2016 12:15 PM - Hexicube

- Status changed from Needs Clarification to Updated

- % Done changed from 0 to 10

Issue occurs on Windows 7 with a fresh 1.2.2 installation, setting the entire Aero folder to read-only causes all wing parts to be missing.

Reverting it fixes the issue, parts being missing after reverting is caused by ModuleManager caching.

Logs contain "access denied" errors. Relevant section of logs: <http://pastebin.com/qWB81twY>

Files are being opened with "File.Open(path, mode)" inside the PartReader class, which uses read/write.

They should instead be using "File.OpenRead(path)", as no writing should occur.

#5 - 01/16/2018 02:25 PM - Ruedii

I think Hexicube has the right lead on this.

It seems that Unity faults on file open of a read-only file being opened with read-write access, instead of on writing it.

It is an issue upstream with Unity, but there is a clear workaround. One could theoretically also fix this with a fault-catch shim on "File.Open" that

retries with "File.OpenRead" and then installs fault stubs where the write routine hooks for the filehandle object would be. so it faults if the program tries to write to the file. It should also log the action.

#6 - 03/20/2018 11:40 AM - Squeelch

- *Severity changed from Critical to Normal*