

# PowerShell - Splitting large log files

☒ Have you ever found an issue with one of your systems and gone digging for the log file, only to find out that the file is too big to open for reading? There are a number of ways you can use in the \*nix world, but for the Windows folks I've created a simple script to let you split the larger file into smaller chunks for easier reading and manipulation using PowerShell.

These are your assumptions for this script to work for you:

- Must be run interactively
- You know the full path and filename of the source file (use double quotation marks to surround the file/path if there are spaces)
- You know the path to the destination (also, use double quotation marks to surround the file/path if there are spaces)
- You have a decent amount of memory (the process loads the entire file...yes...the entire file into RAM while parsing the contents)
- You have some time to let it bake in the background. It's better to wait than to not be able to read the file right?
- You are perfect. There is no error handling (yet). If you make a mistake it will end the way that this gentleman's IT experience did:



What are you doing Dave?

To start we set some initial counters. I'm going old school on this to generate the file names for the output chunk files.

```
# Set the baseline counters
#
# Set the line counter to 0
$linecount = 0
# Set the file counter to 1. This is used for the naming of the log files
$filenumber = 1
```

Now comes the interactive part where we prompt the user (that's you) for the path and filename for the source file. I'm working on making this a GUI file picker, but for now let's just stick with what works. I figure that if you are able to parse a file for content, then I'm assuming you know how to type a file path correctly.

```
# Prompt user for the path
$sourcefilename = Read-Host "What is the full path and name of the log file to split? (e.g. D:mylogfilesmylog.txt)"

# Prompt user for the destination folder to create the chunk files
$destinationfolderpath = Read-Host "What is the path where you want to extract the content? (e.g. d:yourpath)"
```

Because I'm a nice guy, I like to let the user (still you) know that the next process will take a little

while.

```
Write-Host "Please wait while the line count is calculated. This may take a while. No really, it could take a long time."
```

This is where the memory intensive part begins. I'm not an expert programmer, and you may have a better way to do this (feel free to add comments if you have any suggestions). I first read the file to capture a line count so that you can decide how many chunks you want to create. I've also assumed you don't mind a little math.

```
# Find the current line count to present to the user before asking the new line count for chunk files
Get-Content $sourcefilename | Measure-Object | ForEach-Object { $sourcelinecount = $_.Count
}

#Tell the user how large the current file is
Write-Host "Your current file size is $sourcelinecount lines long"
```

Now that you know the size of the original file, the next step is to ask how big the destination files need to be. This will request a number, but as a string, which is converted into an integer assigning a variable and using [int]\$destinationfilesize to convert the string.

```
# Prompt user for the size of the new chunk files
$destinationfilesize = Read-Host "How many lines will be in each new split file?"

# the new size is a string, so we convert to integer and up
# Set the upper boundary (maximum line count to write to each file)
$maxsize = [int]$destinationfilesize
```

Let's tell the user what our results are from the data collection up to this point and then we can begin the process.

```
Write-Host File is $sourcefilename - destination is $destinationfolderpath - new file line count will be $destinationfilesize
```

Here we go! This goes back to programming 101 by using a simple counter and a loop until we reach the threshold of that counter. The output files will be named **splitlog**(some number).**txt** and will each contain the number of lines of text that we've told the script from above.

Final warning! This loads the file into memory which will be pretty resource intensive but the end result is a happy, more easily managed set of files.

```
# The process reads each line of the source file, writes it to the target log file and increments the line counter. When it reaches 100000 (approximately 50 MB of text data)
$content = get-content $sourcefilename | % {
Add-Content $destinationfolderpathsplitleg$filename.txt "$_"
$linecount ++
If ($linecount -eq $maxsize) {
    $filename++
    $linecount = 0
}
}
```

Because we've eaten up lots of memory and processor we will need to free up some of the "garbage" we've left behind. It's effectively a PowerShell poop-n-scoop.

```
# Clean up after your pet  
[gc]::collect()  
[gc]::WaitForPendingFinalizers()
```

And there you have it! Hopefully you find it useful.

Here is the full script - <http://gallery.technet.microsoft.com/PowerShell-Split-large-log-6f2c4da0>



That's much better

---

## [Cadence - Music and Cycling share some technique](#)

In cycling, one of the key techniques to make you a dynamic rider is the mastering of cadence. In music the same holds true. By applying different time signatures and stylistic tweaks such as "cut time" you can add an exciting dynamic to your songs.

As new musicians we are taught fundamentals. Hours and hours of learning to tap out 4/4 rhythm with a metronome to be able to nail down our meter.

tick-tock-tick-tock-tick-  
tock

Now it is time to take things to a new level and excite your listeners with some different rhythms. As a fan of Alt Rock (or grunge as we called it back then) I'll cite Alice In Chains for my example. Let's use Them Bones from the album Dirt to illustrate how a modified time signature can really bring a new edge to your music. AIC guitarist Jerry Cantrell is known for using alternate time signatures, and as a fan of his playing style I've found that it separated the group from many others.

The song opens with a 7/8 riff that immediately pulls you in. The chunking guitar and slightly dragging chord changes under the vocals are incredibly well placed. To add to the dynamic, the chorus is in 4/4 so you can clearly jump in and out with nice stops to delineate the parts.

Another great technique I like to use is "cut time" where you change from 4/4 to 2/2. This takes the natural rhythm to a half speed and when used right it creates what I call a "musical tension" which is kind of like slowly drawing a bow and arrow and then the switch back to 4/4 gives the release.

Since I'm on an Alice in Chains kick today I'll use Sickman from the Dirt album. The song starts with

a strong, quick rhythm followed by a short bridge to transition into the cut time chorus. The chorus closes with a clean stop and then back to the staccato attack of the verse rhythm.

By adding modified time signatures to your arsenal you will add new excitement to your songs and it will bring a great texture to your music. They do say that variety is the spice of life.

---

## [MySQL - Create database, user and privileges](#)



This is a tip for the new MySQL users out there. A common and dangerous practice used by many new developers and admins is to use the root account to access databases. While this certainly gives you the access you need to manage your databases and data from your applications it also elevates privileges far beyond what you really want.

Let's assume that you are creating a database called **mycontent** for an application. You will also want to create a user that has access to just that database so that it isolates the rights to only the content you want to affect.

Decide on a good name for a user. For this example I will use the classic format of databasename\_user so our user will be **mycontent\_user** which is what we will use for our application.

Make sure you choose a complex password as well. It is always best to combine upper and lower case as well as numbers. Be careful with special characters because using some characters like /%\* can cause issues with systems when they pass the characters to a command. We are going to use **7U2JkyGg9TwbNnQ** as our example password.

Launch MySQL from the command line as a user who has privileges to create databases and manage security.

```
mysql -u myadminusername -p
```

Type in your password at the 'Enter password:' prompt

Now we create the database:

```
create database mycontent;
```

Next we create your database user. I am assuming that your database server is also your web server so we will use the @localhost to define the user.

```
grant usage on *.* to mycontent_user@localhost identified by '7U2JkyGg9TwbNnQ';
```

Next step is to apply permissions to the database for your new user:

**grant all privileges on mycontent.\* to mycontent\_user@localhost;**

And finally we check our user is able to log in and access the database. Exit from your MySQL command line and log in again using the new account:

**mysql -u mycontent\_user -p'7U2JkyGg9TwbNnQ' mycontent**

*Welcome to the MySQL monitor. Commands end with ; or g.*

*Your MySQL connection id is 10*

*Server version: 5.5.15 MySQL Community Server (GPL)*

*Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.*

*Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.*

*Type 'help;' or 'h' for help. Type 'c' to clear the current input statement.*

Success!

Now you can use this account/password combination in your application and it will have all of the privileges necessary to manage the database, tables, indexes and content with much less chance of compromising your other databases in your system.

---

## [Cycling and the Ontario Highway Traffic Act](#)

I am a cyclist. It's a simple statement, but you have to understand what it means. I fully support the cycling community by being a law abiding cyclist. That's probably an oxymoron based on what I see on an almost daily basis.



Why underline cycling? The true cycling community is one that supports cycle commuting, recreational cycling, trail usage, amateur and professional racing and along with supporting the riders and organizers of events, we also support drivers, pedestrians and the lawmakers and law enforcement staff who provide safety in a potentially dangerous environment.

While it is admirable for people to cycle to work, or recreationally, it is also a requirement to understand and abide by the simple and easily to understand laws. I fully believe that we should require a licensing system for bicyclists because the entry criteria of "has a bike" isn't enough.

It is of absolute importance that you as a cyclist, and as a driver, fully understand the Highway Traffic Act and how cycling and the law come into play on our Ontario roads. Far too many times I see cyclists and drivers making horrible, dangerous and illegal decisions and it's really infuriating that the standard is set so low for some people.

First assignment: Read this!

<http://www.mto.gov.on.ca/english/pubs/cycling-guide/section5.0.shtml>

A large number of cyclists do these things far too often:

- Run red lights
- Pass cars on the right
- Pass between car lanes to advance position in traffic
- Jump onto the sidewalk to avoid cars

It is very simple. DON'T DO THESE THINGS! If you were on a Motorcycle you would be fully able to do the same moves based on your size and agility yet that seems silly doesn't it?

Hey I can fit so I must be able to ride here

The Toronto Star recently did a great article chronicling the cyclists who breezed through the red light over the course of an hour at a very busy intersection. The result was frightening.

<http://www.thestar.com/news/crime/article/1013091-police-launch-crackdown-on-cyclists-who-don-t-stop>

Hey I get it. The world won't change. There will always be massive numbers of lawless and careless cyclists who ruin it for me and the rest of the proper cyclists out there. Just please do me the favour of not harbouring your anger for hours and then taking it out on me by whipping your steaming hot coffee at me just because I'm on a bike and you can reach me.

I was even pulled over by an off-duty police officer who said that I should not be on the road and that I was "supposed to be on the sidewalk because it was not safe for the cars for me to be on the road". Can someone please tell me how the safety of a person inside a 2000 pound rolling piece of steel armour needs to be protected from a 160 pound guy in spandex riding an 18 pound bicycle?

In this case I asked the fellow to tell me exactly what part of the Highway Traffic Act said that would back up his completely incorrect understanding of the law and safety. So if that unaware gentleman is reading this I'll help him and everyone else here out with this simple excerpt from the HTA:

HTA 147 - Slow moving traffic travel on right side - any vehicle moving slower than the normal traffic speed should drive in the right-hand lane, or as close as practicable to the right edge of the road except when preparing to turn left or when passing another vehicle. ***For cyclists, you must ride far enough out from the curb to maintain a straight line, clear of sewer grates, debris, potholes, and parked car doors. You may occupy any part of a lane when your safety warrants it. Never compromise your safety for the convenience of a motorist behind you.*** Set fine: \$85.00

So let's read through that and understand that the safety of the cyclist is the priority and the "convenience of a motorist behind you" is not. Don't take this as an invitation to ride at a turtle's pace and use up the entire lane just to irk the driver behind you. When I'm rolling along our roads it is at a speed between 30-60 kph so if the simpleton behind me who thinks that he can wipe me out like some character in Grand Theft Auto. This is the real world where when you give me a little

nudge with your car it has a real and devastating effect.

At this point I just try to have enough faith and hope that my family, friends and fellow cyclists can enjoy our sport and our lifestyle safely among the cars and pedestrians.

Know the law. Ride safe. Keep the rubber on the road.