Network Data Plan Usage

* Only files in the queue are files that may be uploaded.
* Data plan information is kept in the “BytesThisBillingCycle.txt” file. It’s in the queue dir.
* Data plan resets on the 20th of each month.
* Data plan limit is 3 GB
* CAMS data plan limit setting is 1 GB.

# **Data Plan Reset**

When the data plan reset date is reached, a scheduled task will run a script, named “**Cams2\_Data\_Plan\_Reset.bat**”, that will reset the data plan usage information, such as the “**data plan month=**” and “**data plan offset=**” values in the **BytesThisBillingCycle.txt** file according to these rules:

* The **NETSTAT -e** command will get the network bytes since reboot in the 5th line of the results. It’s returned as two values, “Received” and “Sent”. Therefore, the value will need to be summed as a 64 bit unsigned integer.
* The network bytes since reboot value will replace the “**data plan offset=**” value in the **BytesThisBillingCycle.txt** file.
* The “**data plan month=**” value is set to the “**bat\_currmm**” var returned by the **GetDate2.bat** script.
* 12 months of data plan usage will be kept in the file, one for each month number. When the “**Cams2\_Data\_Plan\_Reset.bat**” script is run after the “**data plan month=12**”, the **GetDate2.bat** script will return **bat\_currmm=01**. That way, it is cyclical and we can preserve the December data plan usage for analysis and reporting.

# **System Reboot – Reset Network Adapter Bytes**

If the system reboots, a scheduled, task that only runs upon reboot, named “**Cams2\_data\_plan\_reboot.bat**”, will run and perform according to these rules:

1. The “**reboot date string=**” string in the **BytesThisBillingCycle.txt** file will be replaced by the value returned in the second line of a **NET STATISTICS WORK** command. The value will look like this, depending on the locale:
 **Statistics since 11/27/2018 12:27:01 AM**
2. The **NETSTAT -e** command will get the network bytes since reboot in the 5th line of the results. It’s returned as two values, “Received” and “Sent”. Therefore, the value will need to be summed as a 64 bit unsigned integer. The network bytes since reboot value will replace the “**data plan offset=**” value in the **BytesThisBillingCycle.txt** file.
3. The data plan usage bytes for the current billing month will be replaced by:
 **data plan usage + Network bytes since reboot – data plan offset**

**Note:** Any file in the queue that would put us over the MaxBytesPerMonth will remain in the queue until the data plan reset date is reached, at which point the data plan usage value for the current mont will be reset to zero and the older files will be uploaded first. We check the size of a file to upload from the queue before attempting to upload it. If the size of the file, added to the current data plan usage, would put us over the **bat\_CAMS2LIMITS.MaxBytesPerMonth**, then we don’t attempt to upload that specific file. (Subsequent smaller files might be able to pass this check and then successfully upload. )

After each upload, the NETSTAT function is run and the total network bytes are calculated. The data plan usage is calculated as follows:
 **netstat bytes – offset + data plan**

Note, with regard to calculating data plan usage, it almost doesn’t really matter what the size of the upload/download is expected to be (size of the files). We only use the upload file size before uploading to check if we “could” exceed the data plan limits. However, the amount of network bytes used to actually perform an upload or download transfer (due to retries) could be more than double the file size. The NETSTAT value “after” the upload reveals how close to the data plan limits we are. However, if the file size of the upload plus the current plan usage is greater than the plan limits as set in the INI file (bat\_CAMS2LIMITS.MaxBytesPerMonth), then we don’t upload. Remember though, that any small upload could use up so much more than the file size that that upload would push over the data plan limits. Therefore, we need to set the bat\_CAMS2LIMITS.MaxBytesPerMonth value to a value much less than the service provider’s data plan limits. Some files, due to retries, might still get transferred in a way that causes us to exceed the data plan limits. In that case, no further uploads until data plan reset date is reached will be performed.

It’s important to note:

* If the data plan is being exceeded on a regular basis or long before the data plan reset date is reached each month, consider changing the MaxBytesPerMonth value to a very small number. This can effectively disable that automatic uploads altogether. Rather than doing that, it might be better to simply disable the Upload\_Queue scheduled task. Then either re-enable it when you’re ready to download some files or just double-click on upload\_queue\_START.bat.
* Disabling automatic uploads would allow the system to not overrun the MaxBytesPerMonth, thereby leaving the station accessible remotely.
* That way, you can manually upload files. If this is done two to four times a month instead of daily, it could conceivably use less network data than the automatic approach.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Net statistics work DATE** | **Saved BOOTDate** | **Zipsize** | **Netstat Bytes of Uploading ZIP** | **NetstatBytes** | **OFFSET** | **MM=PlanUsage** | **Comments** |
| Dec 20 | Dec 20 | -- | -- | 0 | 0.0 | ? | Data plan reset, for month 12= |
| Jan 10 | “ |  |  | 12.0 | 10.0 | 12=2.0 | Manual adjustment of plan bytes and offset |
| Jan 15 | “ | 0.5 | 0.9 | 12.9 | 10.0 | 12=2.9 |  |
| Jan 20 | Jan 20 | -- | -- | 13.0 | 13.0 | 01=0.0 | Data plan reset, for month 01= |
| Jan 20 | “ | 0.1 | 0.2 | 13.2 | 13.0 | 01=0.2 |  |
| Jan 21 | “ | 0.1 | 0.3 | 13.5 | 13.0 | 01=0.5 |  |
| Jan 22 | “ | 0.4 | 1.0 | 14.5 | 13.0 | 01=1.5 | Over data plan |
| Jan 23 | “ | 0.4 | -- | 14.6 | 13.0 | 01=1.6 | Zip not uploaded – Jan 23 |
| Feb 1 | “ | 0.1 | -- | 14.7 | 13.0 | 01=1.7 | Zip not uploaded – Feb 1 |
| Feb 5 | “ | 0.2 | -- | 14.8 | 13.0 | 01=1.8 | Zip not uploaded – Feb 5 |
| Feb 19 | “ | 0.1 | -- | 14.9 | 13.0 | 01=1.9 | Zip not uploaded – Feb 10 |
| Feb 20 | Feb 20 | -- | -- | 15.0 | 15.0 | 02=0.0 | Data plan reset, for month 02= |
| Feb 20 | “ | 0.1 | 1.6 | 16.6 | 15.0 | 02=1.6 | Zips uploaded for Jan 23, feb 1, feb 5, feb 10. |
| Feb 21 | “ | 0.1 | 0.2 | 16.8 | 15.0 | 02=1.8 |  |
| Feb 22 | Feb 22 | -- | -- | 0.8 | 0.0 | 02=2.6 | Reboot |
| Feb 22 | “ | 0.1 | 0.2 | 0.2 | 0.0 | 02=2.8 |  |
| Feb 22 | Feb 22 | -- | -- | 0.1 | 0.0 | 02=2.9 | Reboot |
| Feb 22 | “ | 0.1 | 0.2 | 0.3 | 0.0 | 02=3.1 |  |
| Feb 23 | “ | 0.1 | 0.2 | 0.5 | 0.0 | 02=3.3 |  |
| Mar 20 | “ | -- | -- | 3.3 | 3.3 | 03=0.0 | Data plan RESET, for month 03= |
| Mar 20 | “ | 0.1 | 0.2 | 3.5 | 3.3 | 03=0.2 |  |
| Mar 21 | “ | 0.4 | 0.5 | 4.0 | 3.3 | 03=0.7 |  |
|  |  |  |  |  |  |  |  |

# **BytesThisBillingCycle.txt**

This file will contain one entry for each month that it encounters. The month may not exist until it us used. For the month entries, the three numbers represent Total Network Bytes, Network Bytes In, Network Bytes Out.

|  |
| --- |
| 01=90546085,30112023,6043406202=3650003,1216667,243333612=7501345,4500807,3000538reboot date string=Statistics since 11/27/2018 12:27:01 AMdata plan month=02data plan offset=1048576 |

# **Reading BytesThisBillingCycle.txt**

## To parse billing file for “reboot date string”:

for /f "usebackq tokens=2\* delims==" %a in (`find /i "reboot date string=" "d:\cams2\_queue\BytesThisBillingCycle.txt"`) do (echo.%a)

## To parse billing file for “data plan offset” value:

for /f "usebackq tokens=2\* delims==" %a in (`find /i "data plan offset=" "d:\cams2\_queue\BytesThisBillingCycle.txt"`) do (echo.%a)

## To parse billing file for “data plan month”:

for /f "usebackq tokens=2\* delims==" %a in (`find /i "data plan month=" "d:\cams2\_queue\BytesThisBillingCycle.txt"`) do (echo.%a)

# **GetNetworkBytes.bat script**

If used with the “/q” option, it does not output to the console. Using this mode, it can be called like a subroutine from other scripts.

Performs the NET STATISTICS WORK and NETSTAT -e and POWERSHELL calculations and returns these environment variables:

bat\_net\_in 2681006112

bat\_net\_out 521956279

bat\_net\_time **Statistics since 11/27/2018 12:27:01 AM**
 A string for comparing whether a reboot has
 occured.

bat\_net\_total 3202962391