

GetStatus Reporting

One of the new updates to the CAMS2 system has been the ability to create status reports and to have a worldwide central location for all the status reports for all systems worldwide.

Introduction

I designed these status reports, mostly for myself, to be able to quickly determine if one of the stations needs me to connect to it and fix something. I have been using them to find all sorts of things and I've been making changes to stations where I find issues. These status reports are also valuable tools for station managers like you all.

These status reports are created at each pass of Upload_Queue.bat (therefore, multiple times per day for each station) and they are stored locally in the "**<cams drive>\cams2_queue**" directory. They are also uploaded to the SETI server "**/incoming/status**" folder. I personally use the "**FTPManager**" app on my iPhone to connect to the ftp server and to view the status reports as they come in to see if there is anything wrong with the systems. So, if the system goes offline, you will still have access to the most recent status report that it was able to upload to SETI for all your systems without having to connect to each system. Therefore, "**/incoming/status**" is a worldwide central location for checking status of CAMS systems.

12 Sections of the status reports:

These status reports are divided into 12 sections:

1. **Date/time, version, and command line** appear at the top. The date/time are in the station's locale. This is so you can quickly tell when the report was generated.
2. **Header** section contains information as read from the camerasites.txt file to help you identify the site. The information includes:
First camera number for the station, Code, Station name, Latitude, Longitude, and Elevation
3. **TIMESTAMP** section contains the UTC and Local times in ISO 8601 standard format of the date/time when the GetStatus report was run on the station it came from. This is so you can compare with the CAMS timestamps, since CAMS timestamps are also in UTC.
4. **Network Data Plan** section – The network data plan statistics restart when a monthly date is reached. This section contains information such as:
 - a. **Last boot date/time** (according to how it's reported by NET STATISTICS WORK command. Windows does not report the time in ISO 8601 format). Windows resets the network adapter counters upon reboot, so this information is required.
 - b. **New Billing Month** – This is the current month at the time the Cams2_Data_Plan_Reset task was run... as parsed from the GetTimestamp2.bat script, which is ISO 8601 date/time.
 - c. **Data Plan Maximum** – Value stored in the Cams2Global.ini file in the [CAMS2LIMITS] section.
 - d. Then there are numbers for **Total, inbound, and outbound** data.

- e. Finally, it shows the **remaining bytes on the data plan**.
5. **Storage** statistics, labeled DRIVE SPACE STATISTICS – shows the **size** and **free** for each of the three cams related drives (Windows, CAMS, Archive).
6. **Archive settings** (from Cams2Global.ini in the [CAMS2ARCHIVE] section. These settings govern when to move the files to Tier-2 (SubmissionFiles) and Tier-3 storage (cams_Archive).
 - a. **MaxDaysToKeep** - Number of days to keep SubmissionFiles sessions until they are culled, zipped, and moved to the “<archive drive>\cams_Archive\archived_SubmissionFiles” location.
Note: These are the largest sets of files. It is ideal to have a <cams drive> large enough to be able to keep 30 days of SubmissionFiles. However, not everyone’s system can accommodate 30 days, so some are set as low as 5-7 days. Contact me if you need to discuss this.
 - b. **MaxDays_Transmitted** – Number of days to keep “<cams drive>\cams2_queue\Transmitted” files. These are the files that your system has transmitted to SETI. These files are not large, but it is a good idea to migrate them to Tier-2 storage.
 - c. **MaxDays_Cal** – Number of days to keep CAL files before archiving them. The default is 365, so we can analyze shifts in the mount through all 4 seasons and temperature changes.
7. **Transmitted section** – “Listing the 10 most recent Transmitted” – These are the 10 most recent sets of files transmitted to SETI. Each should have “.zip” with a matching “.zip.md5.txt” file. Zero byte .zip files, or .zip files with very small size, mean that there is something likely wrong with the capture session. You should look into these sessions and attempt to troubleshoot.
8. **Queue section** – “Listing all the files in the Queue dir” – These are the files now staged to be uploaded during the next pass of upload_queue.bat script, but have not yet been uploaded. Once it is verified that they are transmitted, they are moved to the Transmitted dir and they no longer exist in the Queue dir. It is important that the “.md5.txt” files in the Transmitted dir remain in the Transmitted dir until the files age.
9. **CapturedFiles section** - For each capture board, each **CapturedFiles** capture session is listed, along with their **size**, **number of files**, **number of cameras**, **number of hours of capture** (based on the number of files divided by the amount of time per file per camera) and the **status of the CaptureStats.log** file.
 - a. **NO CAPTURESTATS** means there is no CaptureStats.log file. Interpret this as a CAPTURE FAILED condition. It may have been purposefully manually terminated early.
 - b. **INCOMPLETE** means that the CaptureStats.log file is missing the “CMU graphics cleanup” text that is supposed to exist in a properly finished capture session. This means that the capture session is finished but it got interrupted or that capture is still ongoing.
 - c. **Completed** means that “CMU graphics cleanup” is in the CaptureStats.log file, so we can assume that capture is complete without interruption.
10. **SubmissionFiles section** – In this section, for each board, we list the 10 most recent SubmissionFiles sessions. Each session is divided in two parts. The **main section** and then the **individual camera section**.
 - a. In the **main section**, it shows the **size** and **total number of files** of the entire SubmissionFiles session and all its subdirectories (for those systems with bad/noisy cameras, you will notice that the number of files might be largely different from night to night. That’s because the ArchivedFiles count is also included in this). An 8 camera session (many systems have 16 cameras but with 2 boards, therefore, SubmissionFiles

sessions are for 8 cameras each) would typically have between 25-55 GB and 20,000 to 50,000 files (depending on the time of the year, where winter nights are at least 50% longer than summer nights). We also show the **total number of detections** found in the detect file. Next to that, there are (at present) four status columns for the four main phases of post-capture processing. These phases are: **VALIDATE**, **AUTOCAL**, **DETECT**, AND **TRANSMITTED**. These will show **DONE** or **FAIL** or possibly some other status code to show the status of the post-capture phases for the capture session.

- i. **VALIDATE** – If the VALIDATE phase FAILS, it indicates that there were issues with running the FTP_ValidateFFfiles.exe program. This program shouldn't fail unless something unexpected occurs. There should be a "ValidationLog_<session date/time>_<camera>.txt" file. If that doesn't exist, it means that the validation pass did not run, most likely. If it is empty, then it was likely aborted.
 - ii. **AUTOCAL** – If this FAILs, then it means that either no cameras calibrated or that no AutoCalUpdate.log file exists in the CapturedFiles dir.
 - iii. **DETECT** – If this FAILs, then it means that there is either no detect file (properly named) or something else might be wrong within the file.
 - iv. **TRANSMITTED** – If this FAILs, it means that there is a session that needs to be transmitted, but it has not transmitted. More than likely, it means that there is no matching *.md5.txt hash file for the .zip file. For some reason, the upload to the server failed. There are several reasons this can happen. The most common are:
 1. Your system did not attempt to upload. This can happen if the system reboots before the Upload_Queue.bat script is terminated before it finishes. Check the EventViewer to see if the computer rebooted.
 2. Your system's internet connection was not available at the time the upload was attempted.
 3. Your system's internet was flakey or slow and disconnected one or more times during the attempted upload.
 4. The NASA server's ftp was down or interrupted before or during the upload.
- b. In the **individual camera section**, the information for this comes from the list of cameras that participated in the detect file for the capture session. The information is obtained by reading various files, such as capturestats.log, the detect file, the camerasites.txt file, the CAL files, and more. What's listed for each camera is:
- i. The **camera** that had detection is included in the list but any cameras that did not have any detections are omitted. You should not ignore a condition where a camera does not have any detections.
 - ii. Each camera number is listed along with the **total number of detections** for that camera in the detect file.
 - iii. Next to that is the height and width of the camera FOV in degrees. This is helpful in identifying the lens that is on the camera. Also, sometimes, the FOV can change by one or two degrees. This is condition needs to be corrected and its associated CAL files would need to be deleted and/or recalibrated.
 - iv. Next to that is the image scale in arcminutes per pixel. The image scale should also remain close to the same from session to session. If it changes much, then there is something wrong with the calibration and some troubleshooting needs to be performed, such as recalibrating and deleting recent CAL files.

- v. Next to that is the **CAL file name** being used in the detect file for that camera. The CAL file's date/time stamp section of the CAL file name should be between the session's start date/time and end date/time. The sessions start date/time is easily determined from the date/time portions of the SubmissionFile sessions name. In other words, all the CAL files should indicate that it was done AFTER the start of the session. If it was before the session, then calibration for that camera must have failed. Some stations, like Florida and BeNeLux, are in time zones where a normal capture session could span two UTC dates. For example, today, Florida's start time is 2019_02_06 23_44 UT but their end time is 2019_02_07 11:41 UT.

Note: Cameras with abnormally high detection counts typically point to cameras that have something wrong and some troubleshooting should be performed. High detection counts have been found to be evidence of issues like: noisy camera, bad camera power, bad camera video cable, bad camera video cable connector, incompatible settings (such as NTSC setting for a PAL camera).

Also, number of dropped frames and dropped frames per minute columns are included for each camera.

- vi. Next to that are three columns that provide dropped frames statistics, namely Channel number, Total frames dropped during the session, and average number of dropped frames per minute, if there is an associated CaptureStats.log file for that capture session and if the capture completed. If the dropped frames stats are high for any camera, then the data might very well be useless. This is a condition that needs to be troubleshot and remedied.
11. **CAL files section** – This section lists the 20 most recent CAL files for each board. The reason we chose 20 is so that, for those 16 camera boards (like New Zealand), we can show at least the most recent date, along with a few Calibrations that might have needed to be corrected with a Nudge. What is shown is the Windows **timestamp** (current locale) when the CAL file was created, **Bytes**, and fully qualified **path name**.
 12. **ARCHIVES section** – In this section we list the 10 most recent files in each of the three Archived folders.
 - a. For the **archived_SubmissionFiles**, we show the **timestamp**, **size** of the zip file in MB, and the **full path**.
 - b. Below that, we use the SysInternals "DU.exe" tool to list all the subdirs of the cams_Archive directory. This will show the number of kilobytes of each of the archive section directories. They are not file counts. The number to the left of the right-most comma is the number of megabytes.
 13. **PROCESSES** – this is the final section that shows that state of various important cams-related processes that might have still been running when the GetStatus script was run. Currently, the list includes:
 - a. **LaunchCapture** – If one or more are running, it shows the process ID (PID) of each. There should not be more of these listed than you have number of capture boards. If more instances are shown than you have capture boards, you will need to manually terminate some of them.
 - b. **Capturing** – Lists all the FTP_Capture*.exe processes running. If any are running, it indicates that either capture is still running or that a capture is frozen up. This should

not be more than the number of capture boards you have. If the number listed is higher than the number of capture boards, then LaunchCapture will likely be locked up.

- c. **Validating** – List whether FTP_ValidateFFfiles.exe is running. There should never be more than one.
- d. **AutoCal** – Lists whether FTP_MeteorCal_AutoUpdate.exe is running. There should never be more than one.
- e. **Detecting** – Lists all the FTP_DetectMultipleFF.exe instances are running. There should never be more than one.
- f. **Archiving** – Lists the cmd_archive_cams.exe instance if it is running. There should never be more than one of these running. It starts around 3pm local time so if it's later than that, it might need to be investigated and possibly killed manually.
- g. **Uploading** – Lists the cmd_upload_queue.bat.exe instance. There should never be more than one of these running. It typically starts in the morning an hour or two after capture completes. This is the script that performs the uploads to the SETI server and it also generates these status reports.
- h. **GetStatus** – Lists the cmd_getstatus.bat.exe instance if a status report is being generated.

Attached is a sample status report for ForestHill, CA. It has a single 2 TB hard drive for CAMS that is used also for the cams_Archive. It has 16 cameras with two 8-channel Sensoray cards. The MaxDaysToKeep is set to only 11.



GetStatus_000250_F
H_Foresthill_.txt