Stephen Xu

July 11, 2023

MIS Automated Download Report

**Task Description**

The purpose of this automated Python-based schematic is to download new ERCOT Market Information System (MIS) data daily. The schematic will utilize the ERCOT API to make efficient, secure, and comprehensive requests to ensure the successful retrieval of all required data. Furthermore, the new schematic should have the following advantages over the previous Excel VBA downloader:

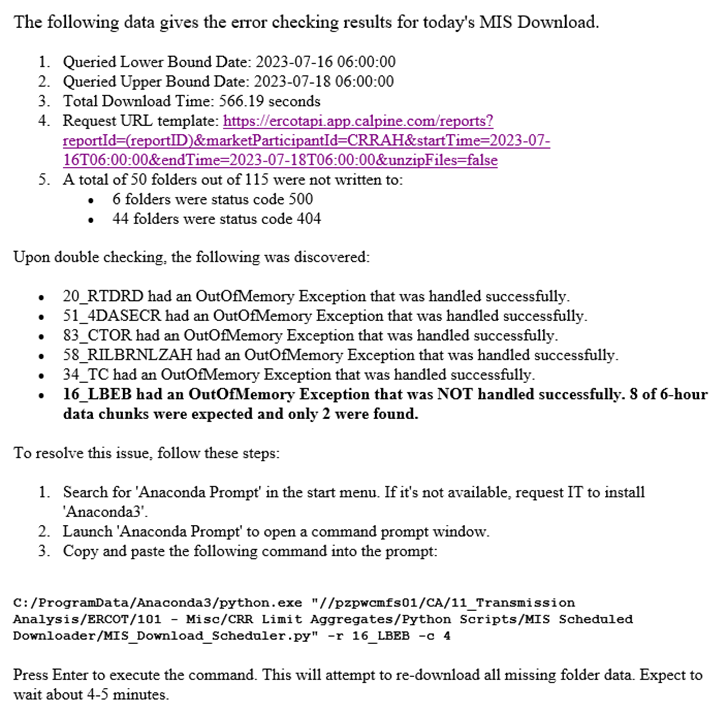
* Detailed Documentation
* Robust Error-Checking Procedures
* Enhanced Error-Handling Capabilities (502 Server Errors, 500 OutOfMemory Exceptions)
* Significant Speedups (7-8 times)

**Solution Design**

1. Script locations: The scripts are located at *\\pzpwcmfs01\CA\11\_Transmission Analysis\ERCOT\101 - Misc\CRR Limit Aggregates\Python Scripts\MIS Scheduled Downloader*
2. Automation: The task runs daily at 6:15 AM on a VDI Task Scheduler, ensuring regular and up-to-date data downloads.
3. Implementation Details
   1. The downloader script utilizes Python's Futures library to perform downloads to each individual subfolder at *\\Pzpwuplancli01\Uplan\ERCOT\MIS 2023* in parallel. This means that multiple downloads are happening at the same time, making the process much faster compared to doing them sequentially. This concurrent approach significantly improves the overall runtime of the script.
   2. Since only one thread can write to each subfolder at a time, there is no potential for data races.
   3. Along the way, the program records all folders that throw a **500 Server Error** exception when querying the API. These errors typically indicate a 'System OutOfMemory' Exception, which means the server ran out of memory while processing a large request. The program records these errors in a log file.
      1. The program handles these folders by querying smaller chunks at time (usually 6-8 hours). This way, the likelihood of running out of memory is much lower.
   4. After the downloading and error handling is complete, the error checker script runs. The script summarizes many daily download details, such as
      1. Lower and upper bounds for date queries
      2. Total download time
      3. Which folders have no available data within the query bounds
      4. Which folders were not successfully/fully downloaded to
   5. If the error checking script identifies any folders that were unsuccessfully downloaded to, it offers an easy solution via the Anaconda Prompt to manually download missing data.
   6. All of the above information is sent in a neat Outlook Email around 6:25 AM.

**Sample Output**

The screenshot below displays a portion of a sample error checking email.



For troubleshooting tips, please refer to the email. If you wish to be added as a recipient of this daily email, navigate to ‘*Error\_Checker.py’* in the script path and edit line **183** by adding your email to the end of the list.