



 **AKUR8**  
**RESERVING**



# Arius Enterprise AEUploader Utility

## Table of Contents

<b>1. DESCRIPTION AND OPERATION.....</b>	<b>1</b>
Installation .....	1
Configuration .....	1
Performing an Upload.....	2
System Requirements & Configuration.....	3
Log Files & Completion Codes.....	3
Optimizing AEUploader with Environment Variables .....	4
<b>2. FREQUENTLY ASKED QUESTIONS .....</b>	<b>5</b>
Why use a command-line utility? Why not a GUI application?.....	5
Why use an API key? Why not regular user credentials?.....	5
What happens if someone obtains my API key? What can they do? .....	5
How can I tell if AEUploader is working? .....	5
How do I check the status of an AEUploader job? .....	5

## 1. Description and Operation

The AEUploader utility is a Windows-based command line executable used to initiate bulk file transfer and database loading of CSV files for Arius Enterprise (AE) into the Triangles on Demand (ToD) database. The utility program leverages Microsoft's AZCopy library to perform user authentication, job configuration, and data transfer. The AEUploader utility connects securely to Azure using several mechanisms:

- HTTPS for all calls to the Azure cloud,
- a Milliman-controlled API key that you obtain as an authenticated user from the Arius Enterprise portal, or
- secure Azure SAS tokens that provide narrow-permissioned time-bound access to data staging locations within Azure.

AEUploader is configured to read from a folder on your local machine and upload all contained CSV files. Accessing network storage is supported but not recommended due to the potential latency of accessing CSVs across your local network.

The AEUploader system supports configurable parallelism so that multiple files can be uploaded at the same time. This reduces upload time at the potential expense of consuming a larger share of available bandwidth on your network. Each time AEUploader runs, it instructs AZCopy to automatically analyze your computer and network setup, then optimize itself for the best possible throughput. If the automatic tuning is not optimal, it is possible to manually tune AZCopy using additional environment variables (see below).

### INSTALLATION

AEUploader does not require a formal installation procedure. To install, unzip the AEUploader zip file into a local folder of your choice.

### CONFIGURATION

All configuration of AEUploader is done within an XML file called AEUploaderMain.exe.config. This file is contained within the zip file along with the AEUploaderMain.exe executable and the AEUploader.bat file used to run the utility. Here are the relevant configuration properties within the config file:

- **TokenUri:** This is the URL to your API server in your organization's dedicated Arius Enterprise Azure subscription and is used to obtain a secure SAS token for CSV upload. If it is not already specified, you can obtain this value from Milliman.
- **ApiKey:** This key is obtained from the Arius Enterprise web portal and used for secure communication to the server at TokenUri. Your Arius Enterprise administrator controls this unique key, and it can be revoked and changed at any time. The APIKey is a unique 128-bit GUID specific to a user account.
- **CsvSource:** Enter the local computer folder location of CSVs to upload. If not specified, it defaults to the current directory of the upload utility.
- **DbName:** This is the target ToD database into which CSV data is to be loaded. The database must exist before import.

- **ValuationDate:** Enter the valuation date to use during import into the target ToD database. This must be in YYYY-MM-DD format.
- **MapperName:** This is the data mapper to use during import into the target ToD database.
- **MaxParallelUploads:** Not currently utilized.
- **UploadOnly:** This flag determines whether target CSV files are merely uploaded to storage (uploadOnly = true) or are both uploaded and then imported into the ToD database (uploadOnly = false). This is defaulted to false.
- **RowsToValidate:** This flag determines how many rows in the CSV file should be checked for data formatting issues before transferring the file to the cloud. If this parameter is not specified, the default value is 10. To check every row in the file, set this parameter to -1. The following items are checked:
  - Does the CSV file header match the schema/mapper?
  - Do values exist for every dimension? Note: dimension fields cannot be empty.
  - Do the date fields have proper date formats?
  - Are the number of columns consistent with the header?
- **SkipValidation:** This true/false flag determines whether or not the validation requested in the RowsToValidate flag should be performed or not. Setting this to true is equivalent to setting RowsToValidate to zero.
- **AZCopyPath:** This field specifies the file path where AZCopy is installed. It defaults to the folder containing the configuration file (where the Zip file installs AZCopy) but can be adjusted if AZCopy is installed elsewhere.
- **AZCopyLogLocation/AzCopyJobPlanLocation:** These fields specify where you would like the logs (or Job Plans) generated by the AZCopy utility to be saved. It defaults to a subfolder of the folder containing AEUploaderMain.exe and Azcopy.exe.
- **SplitSourceFileBeforeUpload:** If set to true, the utility splits the CSV files in the *CSVSource* path into smaller files, with the size specified using the *maxFileSizeInMegaBytes* parameter. This option defaults to false. Any splitting of the files is performed locally before uploading to storage.
- **DestinationOfSplitFiles:** If using the above option, you need to specify a location to place the split files before uploading. This directory must be different than the CSVSource specified above.
- **SplitOnly:** This option allows you to see the file splitting mechanism results without performing any uploads to the portal. SplitOnly defaults to false.
- **MaxFileSizeInMegaBytes:** This option specifies the maximum size of each split file. Note that if your ToD database uses Azure Data Explorer (ADX), this should be set no greater than 2000.

## PERFORMING AN UPLOAD

To run AEUploader, unzip the files to a suitable folder, configure the various properties noted above, and then run AEUploader.bat from the command line. Depending on factors such as the number of files, the size of each file, and your available network bandwidth, the upload may take a few seconds to several minutes, or even hours for gigabytes of data on a relatively slow network.

You can run AEUploader using any other standard means of running Windows EXEs such as batch scripts, Powershell, and automation utilities.



Once your data is uploaded to the secure Azure staging area, AEUploader performs additional processing to extract each CSV file's contents and insert that into the target ToD database (unless the utility is configured to upload only). You can track the progress and view logs of CSV loads within the Arius Enterprise portal (see "How do I check the results of AEUploader?" in the FAQ section below).

AEUploader validates that the CSV files were successfully uploaded to the cloud before kicking off the cloud process to insert the records from the CSV files into the database. If the CSV files were not all transferred to the cloud, it does not start the database insert process.

## SYSTEM REQUIREMENTS & CONFIGURATION

AZCopy can run in many different environments and configurations. It is possible that running AZCopy (and consequently AEUploader) on a Windows computer with limited CPUs and memory may result in unexpected and erratic performance.

As you increase the number of CPU cores, the available memory, and the network bandwidth, AZCopy's performance and behavior improve significantly. You will see material improvements by increasing RAM from 2GB to 16GB, and the number of CPU cores from 4 to 12.

It can be challenging to determine each client's optimal configuration because many factors can contribute to how many simultaneous uploads are instantiated. These can include:

- number and size of files to transfer,
- speed and number of available CPUs, and
- network speed & bandwidth.

It is recommended that an IT professional monitor memory, CPU usage, and network bandwidth while calibrating AEUploader and perform subsequent uploads, if possible. Arriving at the optimal configuration is an iterative process, and the optimal configuration may change over time if the underlying factors change.

Also, note that the optimal settings may change if the machine on which AEUploader resides is part of an auto-scaling virtual environment.

## LOG FILES & COMPLETION CODES

AEUploader generates a log file of the operations and messages displayed on the console while running. The log file is created using the current date and is named bulk-yyyymmdd.log (found in the same directory as the upload utility). If you run the AEUploader multiple times on the same day, the same log file is used and shows information for each run.

AEUploader returns completion and error codes which can be used in the automation process by other applications. The return codes are:

- 0 - Success
- 1 - General error
- 2 - Validation failed
- 3 - Pre-process failed (future, not implemented yet)
- 4 - Upload to cloud failed

5 - Failed to trigger data load

6 - Failed due to data currently being deleted in this database

The AZCopy components of the AEUploader tool also generate log files. When launching AEUploader, take note of the AZCopy log file location, as they are useful for troubleshooting and optimizing, if necessary.

## OPTIMIZING AEUPLOADER WITH ENVIRONMENT VARIABLES

While AEUploader attempts to automatically optimize the uploading process based on an analysis of your computer and network capabilities, it is possible to manually tune AEUploader using additional environment variables. These variables can be set by editing the AEUploader.bat file and adding the corresponding lines below. Note that these settings should only be used if the automatic optimization process does not yield a successful upload. The settings may be useful if the computer you are running the utility on has limited memory.

- **SET AZCOPY\_CONCURRENCY\_VALUE=AUTO:** The batch file used to run AEUploader includes this line, which triggers the automatic scaling process. If you do not wish to allow AEUploader to auto-tune, you may change the value. We recommend values between 6 and 12.
- **SET AZCOPY\_BUFFER\_GB= X:** If the computer running the utility has restricted memory, this may cause unexpected performance issues and possibly crashes. If increasing system memory is not an option, setting this environment variable to a value (such as 0.5 or 0.25, for example) restricts AZCopy's memory usage. This reduces speed but may allow the utility to complete in situations where the lack of this variable results in performance issues.

## 2. Frequently Asked Questions

### Why use a command-line utility? Why not a GUI application?

The batch uploader is intended for a wide range of common upload scenarios. One such common usage is the scheduling of uploads to occur regularly or as the result of files appearing in a specific folder location. These types of unattended execution are difficult or often impossible to accomplish using GUI tools. Command-line EXEs are simple and work with a wide variety of third-party tools and scripting environments.

### Why use an API key? Why not regular user credentials?

User credentials are only appropriate for interactive use cases (a user manually initiating an upload), which is not the only scenario for which the uploader is intended. Unattended or scheduled execution requires authentication schemes like certificate-based authorizations or API keys. Certificates work well but can be challenging to configure appropriately; API keys are an established authentication pattern and have the added advantage of providing maximum control of validity and lifetime to you as an Arius Enterprise administrator.

### What happens if someone obtains my API key? What can they do?

The API key merely allows a user to access the Arius Enterprise API server and obtain an Azure SAS token for uploading data to blob storage. The SAS token is time-restricted to 60 minutes and only allows uploading of data to the secure staging location; it does not permit reads. If you discover an API key has been compromised, you can immediately revoke that key and create a new one if desired, all within the Arius Enterprise portal.

### How can I tell if AEUploader is working?

Within the command prompt (or other utility), there will be refreshes every 2 seconds, highlighting the current status of the AEUploader utility, showing the number of files completed and remaining, as well as any that may have failed. If you run AEUploader and are returned to the command prompt without seeing any of this information, it is likely due to an ongoing delete of data from the ToD portal. Please wait until the delete process is completed, and attempt the upload again.

### How do I check the status of an AEUploader job?

- **Transfer of CSV files to ToD/AE portals:** Examine the bulk loader log file (bulk-yyyyymmdd.log) found within the folder containing AEUploader. It retains information about the transfer of the CSV files to the ToD/Arius Enterprise portal. It does not contain any information about loading these CSV files into the target database.
- **CSV file loaded into a database:** Examine the ToD **View Info** dialog for the associated database to see which CSV files have been successfully inserted into a ToD database.
- **Upload still in progress:** Check the Data Load Log File on the ToD portal to see if any CSV files are still being processed. It is accessed using the download button (⬇️) on the **Databases** tab.