

**Code Functionality**

# About functionality

The code is built for uploading bunch of files from a local folder location to a given blob storage location. Functionalities like copy, move, delete after uploading the files to blob are handled as part of the execution. Every available functionality is driven through an INI configuration file.

# About configuration

INI file is built with 2 sections LOCAL\_INFO & BLOB\_INFO. The configuration details are split between these 2 sections as

## LOCAL\_INFO

* All the information like from where files to be picked up
* What extension files to be picked up
* Actions to be taken on the files after upload
* Supportive information on the actions

## BLOB\_INFO

* Blob connection details
  + Blob name
  + Blob key
  + Container name
* Where to push the files in blob (Virtual blob path)

# Information

* Files from multiple location can be pushed in the same execution with slight change in the configuration file
* Respective destination paths and extensions can be provided for different source folder locations

# About code

## Libraries used

* **azure.storage.blob** – Blob operations (Connect & Upload)
* **coloroma** – Console output log for coloured notations
* **configparser** – Access .ini configuration file
* **datetime, timedelta, time, pathlib, shutil, os, sys** – Utility actions like folder accesses, time calculations

## Output inference

* Status information are provided in Green colour (Success – No issues/failures)
* Debugging information are printed in White colour
* Warning messages are printed in Yellow colour
* Error messages are printed in Red colour
* “All OK” message at the end of the execution denotes the successful code completion

## Code Integrity details

* Program exit code is specified explicitly for successful completion and failure exit. Based on this, the successive task/code can be triggered.
* Program success code = 0
* Program failure code = 1

# Ways to build configuration

## local\_path

* This says files should be picked from which path
* Always use (/) and do not use (\)
* Always end the folder path with slash (/)
* Do not miss out the slash at the end (/)
* Do not provide the file names in the path
* Can provide multiple paths separated by semi colon (;)

## extension\_list

* This says what extension files should be uploaded to blob from the given folder location
* Provide the full extension with dot (.csv) or (.xlsx)
* Multiple extensions can be provided separated by semi colon (;)
* Make sure number of extension provided should be equal to number of folder paths provided in local\_path parameter

## action\_on\_file

* This specifies whether the file should be copied or moved
* 3 options are supported
* copy or move or cut

## destination\_path

* This says files should be moved to which path
* Always use (/) and do not use (\)
* Always end the folder path with slash (/)
* Do not add slashes at the end (/)
* Do not provide the file names in the path
* Can provide multiple paths separated by semi colon (;)

## blob\_name

* Used to connect to the specific blob where the files had to be uploaded

## blob\_key

* Used to connect to the specific blob where the files has to be uploaded
* The configuration files are not encrypted. Please keep the config file safely

## blob\_container

* This specifies in which container the files has to be uploaded

## blob\_path

* This says files should be uploaded to which paths
* Always use (/) and do not use (\)
* Always end the folder paths with slash (/)
* Do not miss out the slash at the end (/)
* Do not provide the file names in the paths
* Do not provide the container name in the paths
* Can provide multiple paths separated by semi colon (;)
* Make sure number of extensions provided should be equal to number of folder paths provided in local\_path parameter

local\_path 🡪 extension\_list 🡪 destination\_path 🡪blob\_path are assumed to have one-to-one relationship. Make sure you provide equal number of entities (Separated by semicolon ;) in all the 4 options

