

# Bulk Import of Email Files

When archiving email files, MailStore only allows archiving of individual email files (PST, MBOX) or directories which contain email files of one user (EML, MSG).

The following describes how to archive all PST-, MBOX-, EML- or MSG-files of all users in a few steps, using a bulk import script written in Python.

The bulk import script expects the email files in folders that are named after the corresponding MailStore users. Such a folder structure can easily be created by another script included in the [scripting package](#).

If you want to archive PST- or MBOX-files of one user only, you can use the [bulkImportPST.bat](#)- and [bulkImportMBOX.bat](#)-scripts from the [scripting package](#); the installation of Python is not required in that case.

## Preparing Python

The provided scripts are written in Python. Python 3.2 or 3.3 is required as well as the [scripting package](#). Python can be downloaded from the Python website [www.python.org](http://www.python.org). The mailstoreapi directory from the scripting package must be moved into the site-packages directory of your Python installation; in Python 3.3 for Windows its default location is C:\Python33\Lib\site-packages. The scripts themselves can be stored in any other location such as "My Documents".

## Preparing the Folder Structure

The scripting package contains the createUserFolders.py script which prepares the folders by fetching a list of existing MailStore Server users and creating folders with the same names in the filesystem. The actual bulk archiving script uses these folder names to assign the enclosed files to the corresponding users in MailStore.

The [MailStore Server Administration API](#) must be enabled in the [MailStore Server Service Configuration](#).

After installing Python, the script can be modified by right-clicking on it, then selecting *DLE*, the supplied Python editor. Adapt the values of USER, PASSWORD, HOST and PORT to your installation.

The rootpath variable contains the path where the folders will be created.

Press F5 to run the script. A security question follows.

## Preparing the Archiving Profile

Before executing the bulk import script, a new archiving profile that corresponds to the email files (PST, MBOX, EML/MSG) to be archived must be created manually in MailStore Server.

The settings in this archiving profile apply to all read files. For instance, if you exclude or include folders or define filters, they will be applied to all files you archive from.

The path you specify in the profile to the PST or MBOX file, or the folder containing EML/MSG files, is irrelevant, as it gets overwritten by the bulk import script. Additionally, sub folders are recognized when archiving EML/MSG files, whereas MailStore headers are always ignored.

### Important Notice

If you want to archive PST files, Microsoft Outlook must be installed on the machine where the archiving profile is run.

Depending on the file type to be archived, the created profiles must be named templateBulkImportEMLMSG, templateBulkImportPST or templateBulkImportMBOX.

## Filling the Folders

For each user, put the email files to be archived into the corresponding folders under the rootpath. A simultaneous import of different file types is not supported. If you want to archive PST and MBOX files, a separate run is necessary for each.

# Archiving

Archiving is done by the `bulkImport.py` script. Before running it must be adapted to your MailStore Server installation by using *IDLE*. Since this script uses the MailStore Client to connect to the server, rather than connecting to the MailStore Server Administration API you have to use slightly other values.

To retrieve the correct parameter values for your installation, open the properties of an existing scheduled task of a MailStore archiving profile and copy the *Run* command line into a text editor of your choice. Alternatively you may pretend creating a new archiving profile for an existing archiving profile, by clicking on *Schedule* and then *Copy CMD*. Now open a text editor and paste the command line to see all parameter values. It is now safe to cancel the creation of the new scheduled task.

For the `MAILSTORECMD` variable, *MailStoreCmdSilent.exe* can be used instead of *MailStoreCmd.exe*. This prevents MailStore Server from displaying empty command line windows while archiving.

The bulk import script requires the encrypted version of the password. This corresponds to the value of the `--pc=` parameter.

Similar to the `createUserFolders.py` script, `rootpath` points to the folder that contains the actual user folders. The `filetype` variable can have one of the following values according to the file type to archive: `pst`, `mbox` or `emlmsg`.

In *IDLE*, press the F5 key to run the script.

# Logging

While the archiving process runs, its progress can be monitored in the status window. The message *exitcode: 0* means that the file or folder has been archived successfully. A different output would show the error. The task logs of the procedures are stored in the `rootpath` folder and are named as follows:

```
Date-Time-User-File.log (PST,MBOX)
```

or

```
Date-Time-User.log (EML/MSG)
```

The table *Recent Results* in MailStore Client gives you an overview of all run archiving profiles. Failed runs can be easily identified by their start time as the file names of the log files begin with the same value.

---

🕒Revision #1

★Created 29 December 2021 17:02:09 by Mahesha Damayanthi

✎Updated 29 December 2021 17:22:27 by Mahesha Damayanthi