

# Backup Files to Amazon S3 with S3Express

---

S3Express is a powerful command line utility for Windows which can list, query, upload and backup files to Amazon S3. This document shows how to setup S3Express to incrementally backup to S3.

*You will need S3Express version 1.4 or newer.*

## Backing Up Files to S3

### Save the S3 Authorization in S3Express

The first step is to enter your Amazon S3 authorization, the access key ID and secret access key, in S3Express. This step only needs to be done once: S3Express will remember it for next runs.

Start S3Express and type:

```
saveauth ACCESS_KEY_ID SECRET_ACCESS_KEY
```

Your **ACCESS\_KEY\_ID** and **SECRET\_ACCESS\_KEY** are displayed in your Amazon S3 account. Once they have been saved in S3Express, the backup can start.

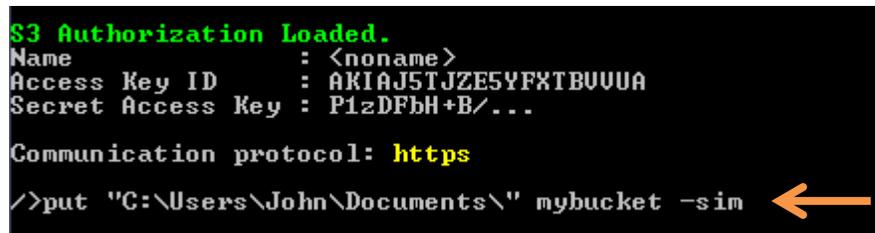
### First Backup

Let's say the local folder to backup is `C:\Users\John\Documents`, the document folder in Windows 7.

Start S3Express and type the following command:

```
put "C:\Users\John\Documents\" mybucket -sim
```

mybucket is the Amazon S3 bucket you want to back up the files to. Like in this picture:



```
S3 Authorization Loaded.  
Name       : <noname>  
Access Key ID   : AKIAJ5TJZE5YFXTBVUUA  
Secret Access Key : P1zDFbH+B/...  
  
Communication protocol: https  
  
>>put "C:\Users\John\Documents\" mybucket -sim ←
```

This command will put all the files that are in `C:\Users\John\Documents\` to mybucket. It will not actually upload the files yet because the `-sim` option was used, which means **simulation**. This option will only preview how many files and the total file size that would be uploaded. To actually start uploading the files, use the same command without `-sim`:

```
put "C:\Users\John\Documents\" mybucket
```

## Including Subfolders

If you also want to include all the subfolders and files of `C:\Users\John\Documents`, use the option `-s` :

```
put "C:\Users\John\Documents\" mybucket -s
```

## Differential Backup

Usually you will only want to upload files that are new or have changed since the last time the backup was run. To achieve this, you can use the option `-onlydiff`. This option instructs S3Express to compare files in `C:\Users\John\Documents\` vs. files in `mybucket` and then only upload those files that are newer in `C:\Users\John\Documents\` plus new files that are not already present in `mybucket`. This will make the backup differential and will speed-up the operation considerably:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff
```

## File Encryption

Uploading files from the local computer to the Amazon S3 servers is performed by S3Express using HTTPs (Hypertext Transfer Protocol Secure). The HTTPs protocol **encrypts all the data flow** between the client (the computer where S3Express is running) and the server (the Amazon S3 servers). This is the same protocol that is generally used to communicate to a bank site when using a web browser. The HTTPs protocol protects against eavesdropping and tampering with and/or forging the contents of the communication. It provides a guarantee that you are communicating with precisely the Amazon S3 servers as well as ensuring that the contents of the communications between S3Express and the Amazon S3 servers (file transfers, file listings, etc.) cannot be read or forged by any third party.

You might also want to keep the files encrypted while they are stored on the Amazon S3 servers. For that you can use the `-e` option of the `put` command:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff -e
```

The `-e` option will enable the Amazon [S3 Server Side Encryption](#).

## Multiple Threads

The upload speed can be significantly increased by using multiple parallel threads. The `-t` option does that:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4
```

`-t:4` indicates to S3Express that we want to upload the files using 4 parallel threads. Usually around 10 threads will achieve maximum performance.

## Restartable Backups

If you have large files to upload and you want to make them restartable in case of an error, you will need to use the option `-mul`, which enables multipart uploads:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul
```

When multipart uploads are enabled with the flag `-mul`, S3Express breaks the files into chunks (by default each chunk is 5MB) and uploads them separately. If the upload of one single chunk fails for whatever reason or if the upload is interrupted, S3Express will restart from the last successful chunk instead of having to re-upload the entire file the next time the backup is run. For more info see [multipart uploads](#).

## Purging Deleted Files

Optionally you can delete from *mybucket* files that are no longer in *C:\Users\John\Documents\*. To do that, you use the option `-purge`:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul -purge
```

`-purge` will delete from *mybucket* files that are not in *C:\Users\John\Documents\*

To preview which files will be deleted, use option `-purge` with `-sim` and `-showdelete`:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul -purge  
-sim -showdelete
```

When purging deleted files, it is a good idea to enable [bucket versioning](#) in Amazon S3. This can be done in the Amazon S3 interface, not in S3Express. When bucket versioning is enabled deleted files are not actually deleted from the bucket but instead they are stored as previous versions and still accessible if needed. Then, using [bucket policies](#), previous versions can be deleted or archived to GLACIER after a specific amount of time. Alternatively, the S3Express `del` command can be used to delete previous versions of files that are older than a certain age.

## Backup to Specific Subfolders

You can back up to a specific subfolder instead of the root of a bucket, e.g.

```
put "C:\Users\John\Documents\" mybucket\subfolder\ -s -onlydiff -e -t:4  
-mul -purge
```

## Backup Only Specific File Types

You can limit the backup to a specific file type:

```
put "C:\Users\John\Documents\*.jpg" mybucket\subfolder\ -s -onlydiff -e
-t:4 -mul -purge
```

or to multiple file types:

```
put "C:\Users\John\Documents\" mybucket\subfolder\ -s -onlydiff -e
-t:4 -mul -purge -include:"*.jpg|*.gif"
```

## Backup Multiple Folders (Multiple Commands)

To backup multiple folders, you can use multiple commands, e.g.

```
put "C:\Users\John\Documents\" mybucket\documents\ -s -onlydiff -e -t:4
-mul -purge
```

```
put "C:\Users\John\Pictures\" mybucket\pictures\ -s -onlydiff -e -t:4
-mul -purge
```

These multiple commands can be saved into a text file, e.g. commands.txt using NotePad. Save the file in the folder c:\temp\ for example and then use S3Express to run all commands that are in the text file, e.g.

```
exec "c:\temp\commands.txt"
```

For more info see [exec command](#).

## Log the Output to a File

You can direct the job output to a file using the -tofile option:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul -purge
-tofile:"log.txt"
```

To append the output to an existing file without erasing its previous content, use +:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul -purge
-tofile:"log.txt+"
```

To have the output going to a different log file each day use the <\*date\*> variable:

```
put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul -purge
-tofile:"log-<*date*>.txt+"
```

## Creating Shortcuts

Instead of typing in a command in S3Express each time you want to backup, you can assign commands to shortcuts to speed up things. Shortcuts are named from c1 to c9. For example type:

```
c1 put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul -purge
    -tofile:"log.txt"
```

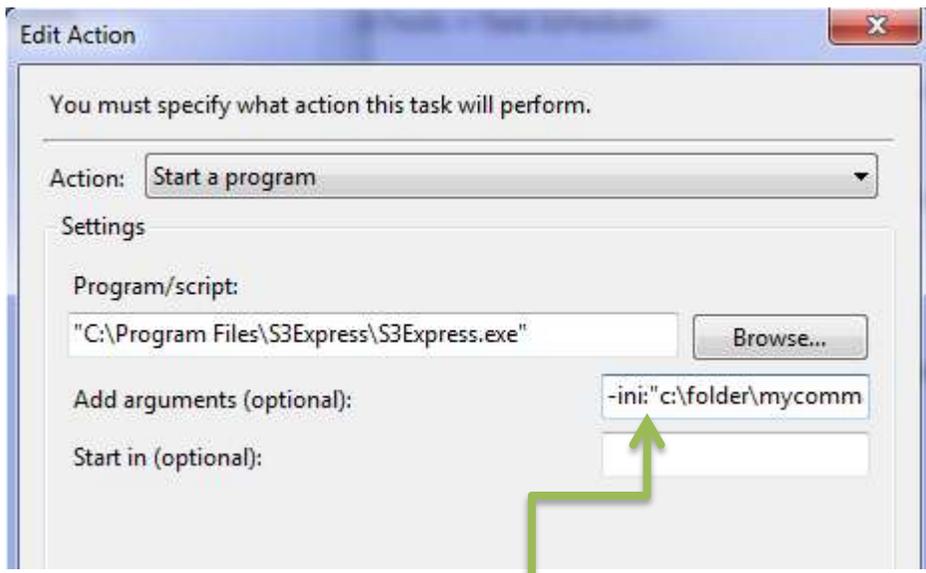
This will assign command `put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul -purge -tofile:"log.txt"` to shortcut **c1**. The next time, you can simply type `c1` to execute `put "C:\Users\John\Documents\" mybucket -s -onlydiff -e -t:4 -mul -purge -tofile:"log.txt"`. There are 9 usable shortcuts, c1, c2, ... till c9. You can show which shortcuts are assigned to which command by typing **cshow** in S3Express.

## Scheduled Backup

The backup can be scheduled by creating a Windows Scheduled Task which will start S3Express automatically. The Windows Task Scheduler is in Accessories -> System Tools -> Task Scheduler.

Start the Task Scheduler and click on "Create Basic Task...". First fill out the name and the schedule that is required (e.g. daily), then in "Start a Program" select S3Express and fill out the 'add arguments' field.

The easiest way is to have all the S3Express backup command(s) you want to run saved in a text file, and then add the text file in the 'arguments' field in Task Scheduler. For example, enter in the 'arguments' field, as shown in the picture below.



```
-ini:"c:\folder\mycommandfile.txt" -exit
```

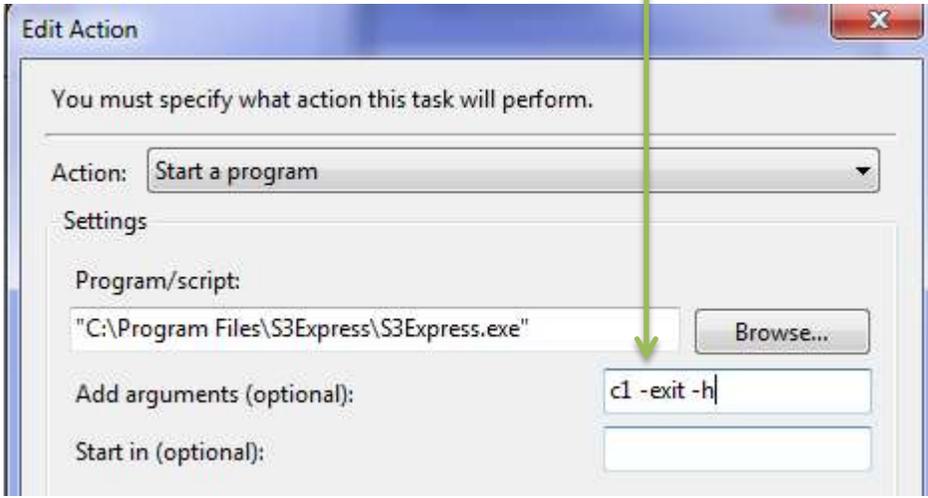
`-ini` instructs S3Express to open the file `c:\folder\mycommandfile.txt` and run the commands contained in the file.  
`-exit` instructs S3Express to exit when done. For more info and command line options see [S3Express command line](#).

If you want S3Express to run invisibly use the -h option:

```
-ini:"c:\folder\mycommandfile.txt" -exit -h
```

You can also use a predefined command shortcut as 'arguments' in the Task Scheduler. See 'Creating Shortcuts' in this document. For example the following will execute the command assigned to shortcut C1 and then exit. See picture below.

```
c1 -exit -h
```



For more info and command line options see [S3Express command line](#).