

S3Express Help

© 2014-2021 TGRMN Software

Help Version: 1.5.2021.11.03 - © 2014-2021 TGRMN Software

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

"Amazon Web Services", the "Powered by Amazon Web Services" logo, "AWS", "Amazon S3" and "Amazon Simple Storage Service" are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Table of Contents

Introduction	1
Overview of All Commands	4
mkbkt (create new bucket)	6
rmbkt (remove bucket)	7
getbktinfo (get bucket information)	8
ls (list objects)	10
cd (change working location)	14
getmeta (show object's metadata)	15
setmeta (set object's metadata)	16
getacl (show object's ACL)	19
setacl (set object's ACL)	20
put (upload files)	24
mkfol (create folder)	32
lsupl (list multipart uploads)	33
rmupl (remove multipart uploads)	34
del (delete objects)	35
copy (copy object)	38
restore (restore objects)	40
Authorization Commands	43
setopt / showopt	45
license (enter license)	49
exec (execute commands from file)	50

Other Commands	52
Filter Condition Syntax (-cond:"FILTER")	55
Command Shortcuts	67
Command Variables	68
Multipart Uploads	70
Scripting via Command Line	71
Exit Codes	73
Redirect Command Output to a File	74
Unicode Support (åäö etc.)	75
Examples	76
FAQ and Knowledge Base	80
How to Buy S3Express	81
How to Enter the License	82
License Agreement	83

1 Introduction

S3Express is a Windows command line utility for Amazon Simple Storage Service S3™.

With S3Express you can access, upload and manage your files on Amazon S3™ using the Windows command line.

S3Express is ideal for scripts, automated incremental backups / uploads and for performing custom queries on Amazon S3™ objects.

S3Express is a very compact program with a very small footprint (the entire program is less than 5MB). It's self-contained in one executable 'S3Express.exe' and it does not require any additional libraries or software to be installed. Simply download and install S3Express and you are ready to go.

Connections to Amazon S3™ are made using secure http (https), which is an encrypted version of HTTP, to protect your files while they are in transit to and from Amazon S3.

S3Express works on all versions of Windows including all Windows Servers.

S3Express is sold, actively supported and maintained by **TGRMN Software**.

Main Features

General

- All S3Express operations are **multithreaded** to achieve maximum speed.
- All S3Express operations are **automatically retried** on connection error (after X seconds and for X times, customizable) to be able to work on less reliable connections.
- All S3Express operations are **interruptible** and then **restartable** at any time simply by pressing the key 'ESC'.
- All S3Express connections to Amazon S3™ are made using **secure http (https)** to protect your files while they are in transit to and from Amazon S3™ servers.
- Multiple [command line variables](#) are supported.
- [Scripting](#) via the command line is supported.
- [Unicode compatible](#), i.e. S3Express supports all alphabet characters in the world.

List Objects

- List objects in one or more S3 buckets and **optionally show metadata and ACL for each object**.
- List objects only in a specified subfolder or recursively list all objects in all subfolders.
- **Include / exclude objects** from the listing based on name, size, metadata, ACL, storage-class, encryption status, etc.
- Filter listing using regular expressions or basic wildcards.
- Show listing **summary only** and **group S3 objects by extension, date, subfolder and more**.
- Optionally include all versions of an object in the listing.
- For example, list all objects with 'cache-control' header not set or 'cache-control' header not equal to a specified value.
- For example, list only all public objects or only all private objects or only objects with a specified ACL.
- For example, list all object whose size is larger than a specified value.

Upload Files

- Upload multiple files and whole directories to Amazon S3.
- Uploads are fully restartable in case of failure.
- Uploads are automatically retried in case of an error (after X seconds and for X times, customizable).
- Optimized parallel file transfers (**multiple threads**) to speed-up uploads.
- Upload files using **multipart uploads** (with correct MD5 value) and multiple threads, so that large uploads can be restarted at any time from where they were left, if interrupted.
- Server-side and/or client-side file encryption supported.
- Keep the existing ACLs and/or metadata when overwriting existing S3 files.
- Throttle **maximum bandwidth** to use in Kilobytes per sec.
- Select files to upload based on name, extension, size, subfolder, time, etc.
- Copy objects instead of re-uploading, if a matching object is found already on S3, so that renaming an object does not require re-uploading.
- Move local files to Amazon S3 after successful upload, based on flexible criteria, e.g. file age, name, size etc.

Incremental Backup to S3

- **Upload only new and changed files.** Using this type of upload you can perform fast, incremental backups: only those files that are new or have changed, compared to the files that are already on S3, are uploaded. If a file is renamed locally, then the corresponding S3 file is copied, not re-uploaded, to save time and bandwidth. Optionally, if a file is deleted locally, the corresponding S3 file can be removed or archived.

Manage Metadata

- Show all or just specific object metadata.
- **Set, reset, replace one or multiple objects' metadata.**
- Preview all operations before proceeding.
- For example, set multiple objects' 'cache-control' header to a certain value.
- For example, apply server-side encryption to existing S3 objects.

Manage ACLs

- Show all or specific object ACLs
- **Set, reset, replace one or multiple objects' ACLs.**
- Preview all operations before proceeding.
- For example, set multiple objects to public access.
- For example, set objects whose name starts with 'R' to private access.
- For example, make sure all objects in a folder or bucket are set to private access.

Delete S3 Files

- Delete one or multiple files from Amazon S3.
- Filter files to delete based on name, extension, size, ACL, metadata, time.

- Stop on first error.
- Preview before deleting.
- Delete previous versions of an object.
- **Multithreaded deletion for maximum speed.**

Copy S3 Files

- Copy Amazon S3 objects.
- Keep metadata and ACLs of copied objects.

Restore S3 Files from GLACIER

- Restore a copy of one or more archived objects from GLACIER to S3.

... and much more!

2 Overview of All Commands

S3Express Commands

Notes:

- For all commands: if objects or path names contain spaces, they must be surrounded with double quotation marks ("), e.g. "c:\folder\file name with spaces.txt"
- All commands support command [variables](#)

Command	Description
Buckets	
mkbkt	Create (make) a new bucket.
rmbkt	Remove a bucket. The bucket must be completely empty without objects or the command will fail.
getbktinfo	Show (get) bucket information.
List Objects	
ls	List objects (i.e. files and folders) in a bucket. Optionally show object's metadata and ACLs.
cd	Change the current S3 working location.
Metadata	
getmeta	Show (get) the metadata associated with an object. Note: use the list command 'ls' to show metadata associated with multiple objects.
setmeta	Set the metadata associated with one or more objects.
ACL	
getacl	Show (get) the access control list (ACL) permissions of an object. Note: use the list command ls to show the ACL permissions of multiple objects.
setacl	Set the access control list (ACL) permissions of one or more objects.
Put (Upload)	
put	Upload files to a S3 bucket.
mkfol	Creates a folder at current working location.
lsupl	List in-progress multipart uploads.
rmupl	Remove in-progress multipart uploads.
Delete	
del	Delete one or more objects from a bucket.
Copy	
copy	Create a copy of an object that is already stored on Amazon S3.
Restore	
restore	Restore objects from Glacier to S3.
Authorization	
saveauth	Save Access Key ID and Secret Access Key in S3Express.
loadauth	Load a previously saved Access Key ID and Secret Access Key in S3Express for

	use.
showauth	Show Access Key ID and Secret Access Key as stored in S3Express.
rmauth	Remove Access Key ID and Secret Access Key from S3Express.
Options	
setopt	Set S3Express options.
showopt	Show S3Express options.
License	
license	Enter a license in S3Express. Entering a license unlocks the S3Express trial.
Exec	
exec	Load and execute a list of commands from a text file.
Shortcuts	
c1, c2, ..., c9	Execute memorized command (shortcut).
Other	
checkupdates	Check for program updates.
md5	Calculate and show MD5 value of a file.
mimetype	Show default mime type used by S3Express for a specific file extension.
OnErrorSkip ResetErrorStatus ShowErrorStatus	Error handling commands. Useful when processing multiple commands with exec or via command line .
pause	Pause for the specified amount of seconds when processing multiple commands.
pwd	Show current local working directory.
Help	
help or h	Show inline help.
htmlhelp	Show help in HTML format.
pdfhelp	Show help in PDF format.
Exit	
q, quit or exit	Exit S3Express.

3 mkbkt (create new bucket)

mkbkt BUCKET_NAME

Create (make) a new bucket.

Parameter	Description	Examples
BUCKET_NAME	Name of the bucket to be created. Required.	mkbkt mybucketname



Restrictions:

- Spaces in bucket names are not allowed.
- Bucket names with upper case letters are not supported.

4 rmbkt (remove bucket)

rmbkt BUCKET_NAME

Remove (delete) an existing bucket.

Parameter	Description	Examples
BUCKET_NAME	Name of the bucket to be removed. Required.	rmbkt mybucketname



Notes:

The bucket to be removed must be completely empty or the command will fail. Remove all objects from a bucket with the [del](#) command before removing a bucket.

5 getbktinfo (get bucket information)

getbktinfo BUCKET_NAME [-acl] [-cors] [-lifecycle] [-policy] [-location] [-logging] [-notification] [-tagging] [-requestPayment] [-versioning] [-website] [-accelerate] [-encryption] [-all] [-orig]

Get bucket information.

Parameter	Description	Examples
BUCKET_NAME	Name of the bucket to be queried. Required.	getbktinfo mybucketname
[-acl]	Shows the access control list (ACL) of the bucket.	getbktinfo mybucketname -acl
[-cors]	Shows the cors configuration information set for the bucket.	getbktinfo mybucketname -cors
[-lifecycle]	Shows the lifecycle configuration information set on the bucket.	getbktinfo mybucketname -lifecycle
[-policy]	Shows the policy of the bucket.	getbktinfo mybucketname -policy
[-location]	Shows the bucket's region.	getbktinfo mybucketname -location
[-logging]	Shows the logging status of the bucket and the permissions users have to view and modify that status.	getbktinfo mybucketname -logging
[-notification]	Shows the notification configuration of the bucket.	getbktinfo mybucketname -notification
[-tagging]	Shows the tag set associated with the bucket.	getbktinfo mybucketname -tagging
[-requestPayment]	Shows the request payment configuration of the bucket.	getbktinfo mybucketname -requestPayment
[-versioning]	Shows the versioning state of the bucket.	getbktinfo mybucketname -versioning getbktinfo mybucketname -versioning -policy (show bucket's policy <i>and</i> versioning)
[-website]	Shows the website configuration associated with the bucket.	getbktinfo mybucketname -website
[-accelerate]	Shows the transfer acceleration state of the bucket.	getbktinfo mybucketname -accelerate
[-encryption]	Shows the encryption state of the bucket.	getbktinfo mybucketname -encryption
[-all]	Shows all of the above for the bucket.	getbktinfo mybucketname -all
[-orig]	Use this parameter to show output in the original S3 format (mostly XML). Do not convert to JSON.	getbktinfo mybucketname -orig getbktinfo mybucketname -versioning -website -orig

**Note:**

It is possible to combine two or more of the parameters above to show multiple items at once, e.g. :
`getbktinfo mybucketname -acl -policy -versioning`

To show all information that applies to a bucket use:

`getbktinfo mybucketname -all`

or just:

`getbktinfo mybucketname`

6 ls (list objects)

ls [BUCKET_NAME]/[FOLDER]/[OBJECT] [-s] [-d] [-od] [-md5] [-r] [-bytes] [-ext] [-sep:SEP] [-showmeta:META] [-showacl:ACL] [-maxkeys:X] [-cond:"FILTER"] [-sum] [-grp:GROUP] [-inclversions] [-onlyprev] [-showverids] [-include:INCL] [-exclude:EXCL] [-rincl:INCL] [-rexcl:EXCL] [-inclenc] [-exclenc] [-inclrr] [-exclrr] [-inclia] [-exclia] [-inclgl] [-exclgl] [-inclle] [-exclle]

List objects (files and folders) in a bucket. Optionally show object's metadata and ACLs.

Parameter	Description	Examples
[BUCKET_NAME]/ [FOLDER]/ [OBJECT]	Name of the bucket, folder, object(s) to list. If not specified, objects in current location are listed. Change the current location with command cd . To specify the parent folder, use '..' Wildcard character can be used (i.e. * and ?) like in Windows <i>dir</i> . If object name have spaces, they must be surrounded by quotation marks (") on the command line.	ls (list all objects in current path. If the current path is the root it will list all buckets) ls mybucket (list all objects in mybucket) ls mybucket/myfolder (list all objects in mybucket/myfolder) ls ../myfolder (list all objects in ../myfolder) ls mybucket/*.txt (list all objects with extension txt in mybucket, in the root folder) ls mybucket/myfolder/*.txt (list all objects with extension txt in mybucket/myfolder/) ls "mybucket/my folder/*.txt" (list all objects with extension txt in mybucket/my folder/, using quotation marks)
-s	Recursive listing, e.g. include all subfolders.	ls -s (list all objects in current path and subfolders. If the current path is the root it will list all buckets and every object in them) ls mybucket -s (list all objects in mybucket and in all subfolders) ls mybucket/myfolder/*.txt (list all objects with extension txt in mybucket/myfolder/ and subfolders)
-d	Include subfolders (=directories) in the listing.	ls mybucket -s -d (list all objects and directories in mybucket and in all subfolders)
-od	Only include subfolders (=directories) in the listing, do not list other objects.	ls mybucket -od (list all folders that are in mybucket) ls mybucket -s -od (list all folders that are in mybucket and in all subfolders)
-md5	Include the object's MD5 value in the listing.	ls mybucket -md5
-r	Regular expression. This flag specifies that [BUCKET_NAME]/[FOLDER]/[OBJECT] must be treated as a regular expression.	ls "mybucket/my folder/*\.txt\.*\.vsn" (list all objects with extension txt or vsn in mybucket/my folder/) ls mybucket/^\r.* (list all objects starting with 'r' in mybucket)
-bytes	Show the object's size in bytes, instead of KB or MB.	ls mybucket -s -bytes
-ext	Extended listing. Metadata and ACLs are shown beneath other object's information.	ls mybucket -ext

-sep:SEP	Use SEP as fields' separator. If not specified, the default separator used is a blank space.	ls mybucket -sep:, (use comma as fields' separator) ls mybucket -sep:"* *" (use '* *' as fields' separator)
-showmeta: "META"	Include specified object metadata in the listing output. Wildcard character can be used (i.e. * and ?). Multiple metadata headers should be separated by . If this flag is not specified, by default, no metadata is shown. Note that showing metadata in the listing output is much slower as each object must be queried separately.	ls mybucket -showmeta:* (include ALL the metadata headers in the output) ls mybucket -showmeta:"cache-control" (include the cache-control header in the output for objects that have it) ls mybucket -showmeta:"cache-control x-amz-server-side-encryption" (include the cache-control and x-amz-server-side-encryption metadata headers in the output for objects that have it) ls mybucket -showmeta:"x-amz-meta-*" (include all the metadata headers that start with x-amz-meta- in the output for objects that have it) ls mybucket -showmeta:* -ext (include ALL the metadata headers in the output in extended format, metadata shown beneath other object's information)
-showacl: "ACL"	Include specified ACL permissions in the listing output. Wildcard character can be used (i.e. * and ?). Multiple ACLs should be separated by . If this flag is not specified, by default, no ACL is shown. Note that showing ACLs in the listing output is much slower as each object must be queried separately.	ls mybucket -showacl:* (include all ACL object permissions in the output) ls mybucket -showacl:allusers (include <i>AllUsers</i> ACL object permissions in the output) ls mybucket -showacl:user (include <i>user</i> ACL object permissions in the output) ls mybucket -showacl:user allusers (include <i>user</i> and <i>alluser</i> ACL object permissions in the output)
-maxkeys:X	Request only X objects per HTTP request.	ls mybucket -maxkeys:10
-cond: "FILTER"	Filter condition. Only include objects in the output matching the specified condition. More info on filter condition syntax and variables.	ls mybucket -s -cond:"extract_value(cache-control,'max-age') > 0" -meta:cache-control (list all objects (recursive) with cache-control:max-age value > 0 in the metadata and include the cache-control header in the output) ls mybucket -s -cond:"size = 0" (list all objects (recursive) of size equal to zero) ls mybucket -s -cond:"name starts_with 'a'" (list all objects (recursive) with name starting with a) ls mybucket -s -cond:"name starts_with 'a' and size > 0" (list all objects (recursive) with name starting with a and size > 0) More info on filter condition syntax and variables
-sum	Show summary only, e.g. total amount of objects and total size, do not list each object separately.	ls mybucket -s -sum (show summary of all objects in mybucket and in all subfolders)
-grp:GROUP	Group objects by GROUP in the output. GROUP can be one of the following values: <i>ym</i> (i.e. -grp:ym) Group objects by year and month. <i>ymd</i> (i.e. -grp:ymd) Group objects	ls mybucket -s -grp:subf (list all objects in mybucket and subfolders and group output by subfolder) ls mybucket -s -grp:ext (list all objects in mybucket and subfolders and group output by object's extension) ls mybucket -s -grp:ymd (list all objects in mybucket and subfolders and group output by year, month and day) ls mybucket -s -grp:ymd -sum (show summary only of all objects in mybucket and subfolders and group summary)

	<p>by year, month and day. <i>ext</i> (i.e. -grp:ext) Group objects by object's extension. <i>subf</i> (i.e. -grp:subf) Group objects by subfolder.</p> <p>GROUP can also be a <i>generic condition</i>, e.g. -grp:s3_sizemb groups object by size in MB, or -grp:cache-control groups object by cache-control value. See Filter Conditions for details.</p>	<p>by year, month and day) ls mybucket -s -grp:cache-control -sum (show summary only of all objects in mybucket and subfolders and group summary by cache-control value) ls mybucket -s -grp:cache-control -sum -cond:"extract_value(cache-control,'max-age') > 0" (show summary only of all objects in mybucket and subfolders which have the cache-control:max-age value larger than 0 and group summary by cache-control value) ls mybucket -s -grp:s3_sizemb -sum (show summary only of all objects in mybucket and subfolders and group summary by object size in megabytes)</p>
-inclversions	Include all object versions (for buckets with object versioning enabled)	ls mybucket -s -inclversions (list all object versions in mybucket and subfolders)
-onlyprev	Include only previous object versions (for buckets with object versioning enabled)	ls mybucket -s -onlyprev (list previous object versions in mybucket and subfolders)
-showverids	Include object version IDs in the in the listing output when object versions are listed (option -inclversions).	ls mybucket -s -inclversions -showverids (list all object versions in mybucket with their version ID)
-include: INCL	Only include objects matching specified mask (Wildcards). Separate multiple masks with " ".	ls mybucket -include:*.jpg (list all jpg files in mybucket) ls mybucket -include:*.jpg *.gif (list all jpg and gif files in mybucket)
-exclude: EXCL	Exclude objects matching specified mask (Wildcards). Separate multiple masks with " ".	ls mybucket -exclude:*.jpg (list all files in mybucket but exclude jpg files) ls mybucket -exclude:*.jpg *.gif (list all files in mybucket but exclude jpg files)
-rinclude: INCL	Only include objects matching specified mask (Regular Expression).	ls mybucket -rinclude:a(x y z)b (list files in mybucket matching axb, ayb and azb) ls mybucket -rinclude:*(.gif bmp jpg) (list files in mybucket matching anything ending with .gif, .bmp or .jpg) ls mybucket -rinclude:"IMGP[0-9]{4}.jpg" (list files in mybucket ending with .jpg , starting with IMG and followed by a four-digit number)
-rexclude: EXCL	Exclude objects matching specified mask (Regular Expression).	ls mybucket -rexclude:[abc] (list all files in mybucket but exclude files containing a, b or c)
-inclenc -exclenc	Include only server-side encrypted files. Exclude server-side encrypted files.	ls mybucket -inclenc (list all files in mybucket that are server-side encrypted) ls mybucket -exclenc (list all files in mybucket that are NOT server-side encrypted)
-inclrr -exclrr	Include only reduced redundancy files. Exclude reduced redundancy files.	ls mybucket -inclrr (list all files in mybucket that are reduced redundancy) ls mybucket -exclrr (list all files in mybucket that are NOT reduced redundancy)
-inclia -exclia	Include only infrequent access files. Exclude infrequent access files.	ls mybucket -inclia (list all files in mybucket that are infrequent access) ls mybucket -exclia (list all files in mybucket that are NOT infrequent access)
-inclgl -exclgl	Include only Glacier files. Exclude Glacier files.	ls mybucket -inclgl (list all files in mybucket that are part of Amazon Glacier)

		ls mybucket -exclgl (list all files in mybucket that are NOT part of Amazon Glacier)
-inclle -exclle	Include only client-side (locally) encrypted files. Exclude only client-side (locally) encrypted files.	ls mybucket -inclle (list all files in mybucket that were locally encrypted) ls mybucket -exclle (list all files in mybucket that were NOT locally encrypted)

**Notes:**

Use quotation marks (") if folder or object names contain blank spaces, e.g. ls "mybucket/my folder/name with a space.txt"

Retry on network error:

The number of retries performed in case of a network error, and the wait time, can be set in the general S3Express options using the command [setopt](#)

7 cd (change working location)

cd [BUCKET_NAME]/[FOLDER]/

Change current S3 working location.

Parameter	Description	Examples
[BUCKET_NAME]/[FOLDER]/	Name of the bucket and folder (optional), that will be set as the new working location.	cd mybucket (working location set to bucket mybucket) cd .. (working location set to parent folder) cd myfolder/ (working location set to subfolder myfolder/) cd "my folder" (working location set to "my folder". Note that surrounding quotation marks are needed to specify names with spaces)

8 getmeta (show object's metadata)

getmeta [BUCKET_NAME]/[FOLDER]/OBJECT [-showmeta:METADATA] [-version:ID]

Shows S3 metadata associated with an object. To show the metadata associated with multiple objects, use the [ls](#) command.

Parameter	Description	Examples
[BUCKET_NAME]/[FOLDER]/OBJECT	Name of the bucket, folder and object to show metadata for.	<p>getmeta file.txt (shows metadata of 'file.txt' at current S3 working location, see cd command for setting working location)</p> <p>getmeta mybucket/folder/file.txt (shows metadata of object 'mybucket/folder/file.txt')</p> <p>getmeta "mybucket/folder/object name with space.txt" (shows metadata of object "mybucket/folder/object name with space.txt". Note that surrounding quotation marks are needed to specify names with spaces)</p> <p>getmeta ../file.txt (shows metadata of object 'file.txt' in parent folder)</p>
-showmeta: "[METADATA]"	Only include specified object metadata in the listing output. Wildcard character can be used (i.e. * and ?). Multiple metadata headers should be separated by . If this flag is not specified, by default, all object's metadata are shown.	<p>getmeta file.txt -showmeta:* (include ALL the metadata headers in the output. This is the default if not specified)</p> <p>getmeta file.txt -showmeta:"cache-control" (include only the cache-control header in the output)</p> <p>getmeta file.txt -showmeta:"cache-control x-amz-server-side-encryption" (include only the cache-control and x-amz-server-side-encryption metadata headers in the output)</p> <p>getmeta file.txt -showmeta:"x-amz-meta-*" (include all the metadata headers that start with x-amz-meta- in the output)</p>
-version:ID	Show the metadata associated with a specific version of the object.	getmeta file.txt -version:12909188 (show metadata of <i>file.txt</i> and version ID: 12909188).



Notes:

To show the metadata associated with multiple objects, use the [ls](#) command.

9 setmeta (set object's metadata)

```
setmeta [BUCKET_NAME]/[FOLDER]/OBJECT [-s] [-r] [-replace] [-meta:META] [-e:+/-] [-rr:+/-] [-ia:+/-] [-sim] [-cond:"FILTER"] [-include:INCL] [-exclude:EXCL] [-rincl:INCL] [-rexcl:EXCL] [-inclenc] [-exclenc] [-inclrr] [-exclrr] [-inclia] [-exclia] [-inclgl] [-exclgl] [-inclle] [-exclle]
```

Set the S3 metadata headers associated with one or multiple objects.

Parameter	Description	Examples
[BUCKET_NAME]/[FOLDER]/OBJECT	Name / path of the object(s) to set metadata for. Wildcard characters are supported by default (* and ?) to match multiple objects. A regular expression can be used too, in that case use the flag -r on the command line, see below.	setmeta mybucket/file -meta:"cache-control:max-age=60" (set header cache-control:max-age=60 to mybucket/file) setmeta mybucket/* -meta:"cache-control:max-age=60" (set header cache-control:max-age=60 to all files in mybucket)
-s	Recursive, e.g. include all objects in all subfolders when processing multiple objects with wildcard characters or regular expression.	setmeta mybucket/* -s -meta:"cache-control:max-age=60" (set header cache-control:max-age=60 to all files in mybucket including in subfolders)
-r	Regular expression. This flag specifies that [BUCKET_NAME]/[FOLDER]/[OBJECT] is a regular expression.	cd mybucket (set working location to mybucket) followed by setacl ^(a.*) (b.*) (c.*) -s -meta:"cache-control:max-age=60" (set header cache-control:max-age=60 to all files starting with a, b or c in mybucket, including files in all subfolders of mybucket)
-replace	Replace the existing metadata headers with the new metadata headers specified with the flag -meta. If -replace is not specified, the new metadata headers specified with the flag -meta will be added to the object(s).	setmeta mybucket/* -meta:"cache-control:max-age=60" -replace (set header cache-control:max-age=60 to all files in mybucket and remove all other metadata)
-meta:META	Metadata headers to be added. Multiple metadata headers should be separated by .	setmeta mybucket/subfolder/file -meta:"cache-control:max-age=60 x-amz-meta-test:yes" (set header cache-control:max-age=60 and x-amz-meta-test:yes to mybucket/subfolder/file)
-e:+/-	-e:+ sets the object S3 server side encryption header 'x-amz-server-side-encryption:AES256'. -e:- removes the object S3 server side encryption header 'x-amz-server-side-encryption:AES256'.	setmeta mybucket/subfolder/file -e:+ (set header 'x-amz-server-side-encryption=AES256' to mybucket/subfolder/file, that will encrypt the file) setmeta mybucket/subfolder/file -e:- (remove header 'x-amz-server-side-encryption=AES256' from mybucket/subfolder/file, that will decrypt the file)

-rr:+/-	<p>-rr:+ sets the object S3 storage class to Reduced Redundancy: 'x-amz-storage-class: REDUCED_REDUNDANCY'.</p> <p>-rr:- removes the object S3 storage class Reduced Redundancy: 'x-amz-storage-class: REDUCED_REDUNDANCY'.</p>	<p>setmeta mybucket/subfolder/file -rr:+ (set header 'x-amz-storage-class=REDUCED_REDUNDANCY' to mybucket/subfolder/file)</p> <p>setmeta mybucket/subfolder/file -rr:- (remove header 'x-amz-storage-class=REDUCED_REDUNDANCY' from mybucket/subfolder/file)</p>
-ia:+/-	<p>-ia:+ sets the object S3 storage class to Infrequent Access: 'x-amz-storage-class: STANDARD_IA'.</p> <p>-ia:- removes the object S3 storage class Infrequent Access: 'x-amz-storage-class: STANDARD_IA'.</p>	<p>setmeta mybucket/subfolder/file -ia:+ (set header 'x-amz-storage-class=STANDARD_IA' to mybucket/subfolder/file)</p> <p>setmeta mybucket/subfolder/file -ia:- (remove header 'x-amz-storage-class=STANDARD_IA' from mybucket/subfolder/file)</p>
-sim	Simulation. Only preview how the metadata would be set and do not actually set the metadata headers yet.	setmeta mybucket/* -meta:"cache-control:max-age=60" -sim (list which files would get the header cache-control: max-age=60 applied to)
-cond: "FILTER"	Filter condition. Only apply the metadata to objects matching the specified condition. More info on filter condition syntax and variables.	setmeta mybucket/* -meta:"cache-control:max-age=60" -cond:"size_mb > 5" (set cache-control:max-age=60 to all files in mybucket that are larger than 5 MB)
-include:INCL	Only apply the metadata to objects matching the specified mask (Wildcards). Separate multiple masks with " ".	setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" - include:"*.exe *.rpt"
-exclude: EXCL	Do not apply the metadata to objects matching the specified mask (Wildcards). Separate multiple masks with " ".	setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" - exclude:"*.exe *.rpt"
-rinclude: INCL	Only apply the metadata to objects matching the specified mask (Regular Expression).	setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" - rinclude:"IMG[0-9]{4}.jpg"
-rexclude: EXCL	Do not apply the metadata to objects matching the specified mask (Regular Expression).	setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" - rexclude:"IMG[0-9]{4}.jpg"
-inclenc -exclenc	Apply the metadata only to server-side encrypted files. Do not apply the metadata to server-side encrypted files.	setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" -inclenc
-inclrr -exclrr	Apply the metadata only to reduced redundancy files. Do not apply the metadata to reduced redundancy files.	setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" -inclrr
-inclia -exclia	Apply the metadata only to infrequent access files. Do not apply the metadata to infrequent access files.	setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" -inclia

<code>-inclgl</code> <code>-exclgl</code>	Apply the metadata only to Glacier files. Do not apply the metadata to Glacier files.	<code>setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" -inclgl</code>
<code>-inclc</code> <code>-exclc</code>	Apply the metadata only to client-side (locally) encrypted files. Do not apply the metadata to client-side (locally) encrypted files.	<code>setmeta mybucket/subfolder/file -meta:"x-amz-meta-test:yes" -inclc</code>

10 getacl (show object's ACL)

getacl [BUCKET_NAME]/[FOLDER]/OBJECT [-showacl:ACL] [-xml] [-version:ID]

Shows S3 ACL associated with an object. To show the ACL of multiple objects, use the [ls](#) command.

Parameter	Description	Examples
[BUCKET_NAME]/[FOLDER]/OBJECT	Name of the bucket, folder and object to show ACL for.	<p>getacl file.txt (shows ACL of 'file.txt' at current S3 working location, see cd command for setting working location)</p> <p>getacl mybucket/folder/file.txt (shows ACL of object 'mybucket/folder/file.txt')</p> <p>getacl "mybucket/folder/object name with space.txt" (shows ACL of object 'mybucket/folder/object name with space.txt'. Note that surrounding quotation marks are needed to specify names with spaces)</p> <p>getacl ../file.txt (shows ACL of object 'file.txt' in parent folder)</p>
-showacl:"ACL"	Only include specified ACL permissions in the output. Wildcard character can be used (i.e. * and ?). Multiple ACLs should be separated by . If this flag is not specified, by default, all object ACL are shown.	<p>getacl object.txt -showacl:* (include all object.txt ACL permissions in the output)</p> <p>getacl object.txt -showacl:allusers (include object.txt <i>AllUsers</i> ACL object permissions in the output)</p> <p>getacl object.txt -showacl:user (include object.txt <i>user</i> ACL object permissions in the output)</p> <p>getacl object.txt -showacl:user allusers (include object.txt <i>user</i> and <i>alluser</i> ACL object permissions in the output)</p>
-xml	Output ACL in raw XML format.	getacl object.txt -xml
-version:ID	Show the ACL associated with a specific version of the object.	getacl object.txt -version:23444411 (show ACL of <i>object.txt</i> , object version ID 23444411)



Notes:

To show the ACL of multiple objects, use the [ls](#) command.

11 setacl (set object's ACL)

```
setacl [BUCKET_NAME]/[FOLDER]/OBJECT [-s] [-r] [-cacl:
CANNED_ACL] [-grant-read:"GRANTEE"] [-grant-write:"GRANTEE"] [-
grant-full-control:"GRANTEE"] [-grant-read-acp:"GRANTEE"] [-grant-
write-acp:"GRANTEE"] [-sim] [-cond:"FILTER"] [-include:INCL] [-
exclude:EXCL] [-rincl:INCL] [-rexcl:EXCL] [-inclenc] [-exclenc]
[-inclrr] [-exclrr] [-inclia] [-exclia] [-inclgl] [-exclgl] [-inclle] [-
exclle]
```

Set the S3 ACL for one or multiple objects.

Parameter	Description	Examples
[BUCKET_NAME]/ [FOLDER]/ OBJECT	Name / path of the object(s) to set the ACL for. Wildcard characters are supported by default (* and ?) to match multiple objects. A regular expression can be used too, in that case use the flag -r on the command line, see below.	setacl mybucket/file -cacl:private (set canned ACL 'private' to mybucket/file) setacl mybucket/* -cacl:public-read (set canned ACL 'public-read' to all files in mybucket) setacl mybucket/*.txt -s -cacl:public-read-write (set canned ACL 'public-read-write' to all txt files in mybucket, including in subfolders of mybucket)
-s	Recursive, e.g. include all subfolders when processing multiple objects with wildcard characters or regular expression.	setacl mybucket/*.txt -s -cacl:public-read-write (set canned ACL 'public-read-write' to all txt files in mybucket, including in subfolders of mybucket)
-r	Regular expression. This flag specifies that [BUCKET_NAME]/[FOLDER]/[FILE] is a regular expression.	cd mybucket (set working location to mybucket) <i>followed by</i> setacl ^(a.*)(b.*)(c.*) -s -cacl:public-read (set canned ACL 'public-read' to all files starting with a, b or c in mybucket, including files in subfolders of mybucket)
-cacl: CANNED_ACL	Set canned ACL. Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. Valid Values for CANNED_ACL: private (Owner gets FULL CONTROL. No one else has access rights, this is the default for an object) public-read (Owner gets FULL CONTROL. The AllUsers group, that is everyone, gets READ access)	setacl mybucket/*.jpg -s -cacl:private (set canned ACL 'private' to all jpg files in mybucket, including in subfolders of mybucket)

	<p>public-read-write (Owner gets FULL CONTROL. The AllUsers group, that is everyone, gets READ and WRITE access)</p> <p>authenticated-read (Owner gets FULL CONTROL. The AuthenticatedUsers group, that is all Amazon AWS accounts, gets READ access.)</p> <p>bucket-owner-read (Object owner gets FULL CONTROL. Bucket owner gets READ access)</p> <p>bucket-owner-full-control (Both the object owner and the bucket owner get FULL CONTROL over the object)</p> <p>Note: You can specify only one of these canned ACLs in your request.</p>	
-grant-read:"GRANTEE"	Allows grantee to read the object data and its metadata. See how to specify one or more grantees below.	setacl mybucket/* -grant-read:"emailAddress=xyz@amazon.com, emailAddress=abc@amazon.com"
-grant-write:"GRANTEE"	Allows grantee to write the object data and its metadata. See how to specify one or more grantees below.	setacl mybucket/* -grant-write:"emailAddress=xyz@amazon.com, emailAddress=abc@amazon.com"
-grant-full-control:"GRANTEE"	Allows grantee the read, write, read_acp and write_acp permissions on the object, that is full control.	setacl mybucket/subfolder/* -grant-full-control:"uri=http://acs.amazonaws.com/groups/global/AllUsers"
-grant-read-acp:"GRANTEE"	Allows grantee to read the object ACL. See how to specify one or more grantees below.	setacl mybucket/subfolder/* -grant-read-acp:"uri=http://acs.amazonaws.com/groups/global/AllUsers"
-grant-write-acp:"GRANTEE"	Allows grantee to write the object ACL. See how to specify one or more grantees below.	setacl mybucket/subfolder/* -grant-write-acp:"uri=http://acs.amazonaws.com/groups/global/AllUsers"
-sim	Only preview how the ACL would be set, do not actually set the ACL for objects.	setacl mybucket/*.jpg -s -cacl:private -sim (simulate setting canned ACL 'private' to all jpg files in mybucket, including in subfolders of mybucket, without actually setting yet, i.e. preview only)
-cond: "FILTER"	Filter condition. Only apply the permissions to objects matching the specified condition. More info on filter condition syntax and variables.	<p>setacl mybucket -s -cacl:private -cond:"s3_sizeMB > 5" (set canned ACL 'private' to all files in mybucket and subfolders that are larger than 5 Megabytes)</p> <p>setacl mybucket -s -cacl:private -cond:"to_lower(s3_extension) = '.exe'" (set canned ACL 'private' to all files in mybucket and subfolders that have extension .exe, case insensitive)</p>
-include:INCL	Only apply the permissions to	setacl mybucket -s -cacl:private -include:*.jpg *.gif (set

	objects matching the specified mask (Wildcards). Separate multiple masks with " ".	canned ACL 'private' to all files in mybucket and subfolders that have extension .jpg or .gif)
-exclude: EXCL	Do not apply the permissions to objects matching the specified mask (Wildcards). Separate multiple masks with " ".	setacl mybucket -s -cacl:private -exclude:*.jpg *.gif *.png (set canned ACL 'private' to all files in mybucket and subfolders, excluding files that have extension .jpg or .gif or *.png)
-rincl: INCL	Only apply the permissions to objects matching the specified mask (Regular Expression).	setacl mybucket -s -cacl:private -rincl:a(x y z)b (set canned ACL 'private' to all files in mybucket and subfolders whose name is matching axb, ayb and azb)
-rexcl: EXCL	Do not apply the permissions to objects matching the specified mask (Regular Expression).	setacl mybucket -s -cacl:private -rexcl:a(x y z)b (set canned ACL 'private' to all files in mybucket and subfolders, excluding files whose name is matching axb, ayb and azb)
-inclenc -exclenc	Apply the permissions only to server-side encrypted files. Do not apply the permissions to server-side encrypted files.	setacl mybucket -s -cacl:private -inclenc (set canned ACL 'private' to all files in mybucket and subfolders that are server-side encrypted)
-inclrr -exclrr	Apply the permissions only to reduced redundancy files. Do not apply the permissions to reduced redundancy files.	setacl mybucket -s -cacl:private -inclrr (set canned ACL 'private' to all files in mybucket and subfolders that have storage class 'reduced redundancy')
-inclia -exclia	Apply the permissions only to infrequent access files. Do not apply the permissions to infrequent access files.	setacl mybucket -s -cacl:private -inclia (set canned ACL 'private' to all files in mybucket and subfolders that have storage class 'infrequent access')
-inclgl -exclgl	Apply the permissions only to Glacier files. Do not apply the permissions to Glacier files.	setacl mybucket -s -cacl:private -inclgl (set canned ACL 'private' to all files in mybucket and subfolders that have storage class 'Glacier')
-inclle -exclle	Apply the permissions only to client-side (locally) encrypted files. Do not apply the permissions to client-side (locally) encrypted files.	setacl mybucket -s -cacl:private -inclle (set canned ACL 'private' to all files in mybucket and subfolders that are client-side encrypted)



How to specify a **GRANTEE**:

You specify each grantee as a *type=value* pair, where the *type* can be one of the following:

emailAddress - if value specified is the email address of an AWS account

id - if value specified is the canonical user ID of an AWS account

uri - if granting permission to a predefined group.

Multiple grantee must be separated by a comma.

For example, the following `-grant-read` grants read object data and its metadata permission to the AWS accounts identified by their email addresses:

```
-grant-read:"emailAddress=xyz@amazon.com, emailAddress=abc@amazon.com"
```

The following `-grant-full-control` grants full control to everyone:

```
-grant-full-control:"uri=http://acs.amazonaws.com/groups/global/AllUsers"
```

Refer to the Amazon S3 documentation for a full list of uri supported.

12 put (upload files)

```
put LOCAL_FILES [BUCKET_NAME]/[FOLDER]/[OBJECT] [-s] [-t:
THREADS] [-mul:PARTSIZE] [-maxb:MAXB] [-cacl:CANNED_ACL] [-
meta:METADATA] [-mime:MIMETYPE] [-e] [-le] [-rr] [-ia] [-r] [-
cond:"FILTER"] [-nomulmd5] [-nomd5existcheck] [-nobucketlisting]
[-keep:KEEP] [-onlydiff] [-onlynewer] [-onlynew] [-onlyexisting] [-
purge] [-purgeabort:X] [-move] [-localdelete:"COND"] [-include:
INCL] [-exclude:EXCL] [-rincl:INCL] [-rexcl:EXCL] [-sim] [-
showfiles] [-showdelete] [-showlocaldelete] [-showexcl] [-
noautostatus] [-minoutput] [-stoponerror] [-optimize] [-accelerate]
```

Upload one or multiple files (=objects) to a S3 bucket. If an identical file (i.e. same MD5 value) is already stored on Amazon S3, the file is copied, not uploaded, to save bandwidth.

Parameter	Description	Examples
LOCAL_FILES	Name / path of the local file(s) to upload. Wildcard characters are supported by default (* and ?) to match multiple objects. A regular expression can be used too, in that case use the flag -r on the command line, see below.	put c:\folder\ mybucket (upload all files in c:\folder\ to mybucket) put c:\folder\file.txt mybucket (upload file c:\folder\file.txt to mybucket) put c:\folder*.txt mybucket (upload files *.txt in c:\folder\ to mybucket)
[BUCKET_NAME]/ [FOLDER]/ [OBJECT]	Name of S3 bucket, folder (optional) and object (optional) to upload files to. This is relative to the current S3 working location.	put c:\folder\file.txt mybucket/subfolder/ (upload file c:\folder\file.txt to mybucket/subfolder) put c:\folder*.txt mybucket/subfolder/ (upload files *.txt in c:\folder\ to mybucket/subfolder)
-s	Recursive, upload local files that are in subfolders too. The subfolder structure is replicated while uploading.	put c:\folder\ mybucket -s (upload all files in c:\folder\ and subfolders of c:\folder\ to mybucket. The subfolder structure is replicated in mybucket)
-t:THREADS	Specify the number of concurrent, parallel threads used to upload files to S3. By default only 1 thread is used.	put c:\folder\ mybucket -s -t:4 (upload all files in c:\folder\ and subfolders of c:\folder\ to mybucket using 4 parallel threads)
-mul: PARTSIZE	Use Amazon S3 multipart uploads to upload the files. The PARTSIZE value is optional and can be used to specify the size of each upload part to use, in Megabytes. The minimum upload part size is 5MB and that is also the default size used if PARTSIZE is not specified. Max size is 1000 Megabytes. The -mul flag is required when	put c:\folder\ mybucket -s -t:4 -mul (upload all files in c:\folder\ and subfolders of c:\folder\ to mybucket using 4 parallel threads and multipart uploads) put c:\folder\ mybucket -s -t:4 -mul:50 (upload all files in c:\folder\ and subfolders of c:\folder\ to mybucket using 4 parallel threads and multipart uploads. Use an upload part size of 50 Megabytes)

	uploading files larger than 5GB and it is recommended when uploading files larger than 200MB.	
-maxb:MAXB	Specify maximum bandwidth to use in KiloBytes/sec. For example -maxb:100 instructs S3Express to use max 100KB/sec to upload.	put c:\folder\ mybucket -maxb:50 (upload all files in c:\folder\ to mybucket, throttle bandwidth to 50KB/s)
-cacl: CANNED_ACL	<p>Set canned ACL of uploaded files. Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions.</p> <p>Valid Values for CANNED_ACL:</p> <p>private (Owner gets FULL CONTROL. No one else has access rights, this is the default for an object)</p> <p>public-read (Owner gets FULL CONTROL. The AllUsers group, that is everyone, gets READ access)</p> <p>public-read-write (Owner gets FULL CONTROL. The AllUsers group, that is everyone, gets READ and WRITE access)</p> <p>authenticated-read (Owner gets FULL CONTROL. The AuthenticatedUsers group, that is all Amazon AWS accounts, gets READ access.)</p> <p>bucket-owner-read (Object owner gets FULL CONTROL. Bucket owner gets READ access)</p> <p>bucket-owner-full-control (Both the object owner and the bucket owner get FULL CONTROL over the object)</p> <p>Note: You can specify only one of these canned ACLs in your request.</p>	put c:\folder\ mybucket -cacl:public-read (upload all files in c:\folder\ to mybucket and make all uploaded files 'public-read')
-meta:META	Metadata headers to be added to the uploaded files. Multiple metadata headers should be separated by .	put c:\folder\ mybucket -meta:"cache-control:max-age=60" (upload all files in c:\folder\ to mybucket and set metadata header 'cache-control' to max-age=60 for all uploaded files)
-mime: MIMETYPE	Specify the MIME type to assign to uploaded files. By default S3Express assigns standard MIME types (HTTP header	put c:\folder\ mybucket -mime:"mymime" (upload all files in c:\folder\ to mybucket and set mime header 'Content-Type' to 'mymime' for all uploaded files, overriding the default values)

	"Content-Type"). You can override these default values for uploaded files by using the flag -mime.	
-e	Apply Amazon S3 Server Side Encryption to uploaded files.	put c:\folder\ mybucket -e (upload all files in c:\folder\ to mybucket and apply server side encryption for all uploaded files)
-le	<p>Apply local encryption before uploading files and then upload the encrypted files.</p> <p>Local encryption is performed using the open-source file encryption program AEScrypt, which can be downloaded from www.aescrypt.com</p> <p>Download the command line version of AEScrypt for Windows and save the file aescrypt.exe in the same folder where S3Express.exe is.</p> <p>To provide an encryption password, use the command setopt, with flag <i>-clientencpwd</i>.</p> <p>To provide an encryption password hint, use the command setopt, with flag <i>-clientencpwdhint</i>. If a password hint is specified, it is then added to the metadata of each encrypted file. The metadata header containing the password hint is 'x-amz-meta-s3xpress-encrypted-pwd-hint'.</p> <p>The original MD5 of the unencrypted file is added to the object metadata in the header 'x-amz-meta-s3xpress-encrypted-orig-md5'.</p> <p>For each encrypted object also the metadata header 'x-amz-meta-s3xpress-encrypted:aescrypt.exe' is added.</p> <p>Alternative encryption programs, such as 7zip or other custom programs, can be specified using the command setopt with option <i>-clientencprogram</i>.</p>	<p>put c:\folder\ mybucket -le (upload all files in c:\folder\ to mybucket. Before uploading, apply client side local encryption using the program AEScrypt. Note that to provide an encryption password the command setopt -<i>clientencpwd</i> must be used first)</p> <p>The flags -e and -le can be combined, e.g.:</p> <p>put c:\folder\ mybucket -e -le (upload all files in c:\folder\ to mybucket. Before uploading, apply client side local encryption using the program AEScrypt. Also apply server side encryption for all uploaded files)</p>
-rr	Set S3 storage class to "Reduced Redundancy" for uploaded files (REDUCED_REDUNDANCY).	put c:\folder\ mybucket -rr (upload all files in c:\folder\ to mybucket and set Storage Class to 'REDUCED_REDUNDANCY' for all uploaded files)
-ia	Set S3 storage class to "Infrequent Access" for uploaded	put c:\folder\ mybucket -ia (upload all files in c:\folder\ to mybucket and set Storage Class to 'STANDARD_IA' for all

	files (STANDARD_IA).	uploaded files)
-r	Regular expression. This flag specifies that LOCAL_FILES is a regular expression <u>operating on the current folder</u> . If you want to apply a regular expression to a folder other than the current folder, use the -cond:FILTER condition or even easier the flag -rinclude or -rexclude, see below.	put ^(a.*) (b.*) (c.*) mybucket -r (upload files starting with a, b, or c in the <u>current folder</u> to mybucket. The -r flag only operates on the current local folder)
-cond:FILTER	Filter condition. Only upload files matching the specified condition. More info on filter condition syntax and variables.	put c:\folder\ mybucket -cond:"size <> 0" (upload non-empty files from c:\folder\ to mybucket)
-nomulmd5	Do not recalculate MD5 for files uploaded in multipart mode (see put flag -mul above). When uploading files in multipart mode (-mul), S3Express will force MD5 recalculation for files smaller than 1GB at the end of the upload. Use this flag to disable MD5 recalculation. If needed, the 1GB limit can be changed in the Windows Registry.	put c:\folder\ mybucket -mul -nomulmd5 (upload all files in c:\folder\ to mybucket using multipart uploads and do not force recalculation of MD5 values)
-nomd5existcheck	By default, if S3Express finds an identical file (i.e. same MD5 value) that is already stored on Amazon S3, then that file is replicated and not uploaded again, to save time and bandwidth. This happens only for files <200MB. S3Express will show which files are copied (=duplicated) instead of uploaded. This functionality can be disabled using this flag -nomd5existcheck.	put c:\folder\ mybucket -nomd5existcheck
-nobucketlisting	This option forces S3Express not to list the remote S3 bucket. Instead of listing the remote S3 bucket before the put operation starts, S3Express will check <i>file by file</i> if a local file needs to be uploaded. This option can be quite slow, but it is faster when a few files are to be uploaded to a large S3 bucket that already has lot of files in it. This option is not compatible with the options -purge, -nobucketlisting and -le, an error will be given in that case.	put c:\folder\ mybucket -onlydiff -nobucketlisting
-keep:KEEP	If the files to be uploaded have a matching file already in S3 that	put c:\folder\ mybucket -keep (upload all files in c:\folder\ to mybucket and keep metadata and ACL for S3

	<p>will be overwritten, keep the existing metadata and/or ACL. -keep:acl keeps the existing ACL -keep:meta keeps the existing metadata -keep keeps both, metadata and ACL.</p>	files that will be overwritten)
-onlydiff	<p>Only upload files that are <i>different</i> compared to the matching files that are already on S3. Different files are files that have the same path and the same name but a different MD5 value. Different files are also files that are not yet uploaded to S3. So using the '-onlydiff' flag uploads files that are not yet on S3 plus all the files whose content has changed compared to the files already on S3.</p> <p>This flag is equivalent to using -cond:"etag != s3_etag".</p> <p>Note that if the upload part size (-mul) is changed in between uploads, then a file may be re-uploaded even if it is already on S3. The -onlydiff functionality only works when -mul size is kept the same between uploads or -mul is not used.</p> <p>Running twice the same <i>put command</i> with the flag -onlydiff is a good way to verify that all files have been uploaded correctly: all MD5 values should already match, unless local files have been changed since last upload.</p>	<p>put c:\folder\ mybucket -onlydiff (upload files in c:\folder\ to mybucket only if they are <i>different</i> compared to the matching file that is already on S3. Different files are files that have the same path and the same name but a different MD5 value. Files that have already a corresponding file with matching MD5, will not be uploaded)</p> <p>put c:\folder\ mybucket -onlydiff -nobucketlisting (do the same as above but without listing the S3 bucket)</p>
-onlynewer	<p>Only upload files that are <i>newer</i> compared to the matching files that are already on S3. Newer files are files that have the same path and the same name but a newer modified time. Newer files are also files that are not yet uploaded to S3. So using the '-onlynewer' flag uploads files that are not yet on S3 plus all the files whose timestamp is newer compared to files already on S3.</p> <p>This flag is equivalent to using -cond:"timestamp > s3_timestamp".</p> <p>Note that <i>-onlynewer</i> is faster than <i>-onlydiff</i>, because the MD5</p>	put c:\folder\ mybucket -onlynewer (upload files in c:\folder\ to mybucket only if they are <i>newer</i> compared to the matching file that is already on S3. <i>Newer</i> files are files that have the same path and the same name but a newer modified time)

	value of local files does not need to be calculated when using <code>-onlynewer</code> .	
<code>-onlynew</code>	Only upload files that are <i>new</i> , that is not yet on S3. Using <code>-onlynew</code> only uploads files that are not yet on S3. This is equivalent to using <code>-cond:"s3_etag = ""</code> .	<code>put c:\folder\ mybucket -onlynew</code> (upload files in <code>c:\folder\</code> to mybucket only if they are <i>new</i> , that is, they do not have a matching file that is already on S3)
<code>-onlyexisting</code>	Only upload files that are already <i>existing</i> on S3. Using <code>-onlyexisting</code> only uploads files that already have a corresponding matching file with same name and path on S3. This is equivalent to using <code>-cond:"s3_etag <> ""</code> .	<code>put c:\folder\ mybucket -onlyexisting</code> (upload files in <code>c:\folder\</code> to mybucket only if they are already <i>existing on S3</i> , that is, they already have a matching file on S3)
<code>-purge</code>	Delete S3 files that no longer exist locally.	<code>put c:\folder\ mybucket -onlydiff -purge</code> (upload files in <code>c:\folder\</code> to mybucket only if they are <i>different</i> compared to the matching file that is already on S3. Delete files in mybucket that are not in <code>c:\folder\</code>)
<code>-purgeabort: X</code>	Abort the purge operation if more than X S3 files would be deleted. X can be: - The number of files - ALL (it specifies to abort if all files in the S3 bucket would be deleted, this is the default behavior) - NEVER (it specifies to never abort purge)	<code>put c:\folder\ mybucket -onlydiff -purge -purgeabort:100</code> (upload files in <code>c:\folder\</code> to mybucket only if they are <i>different</i> compared to the matching file that is already on S3. Delete files in mybucket that are not in <code>c:\folder\</code> . Do not purge if more than 100 S3 files would be deleted) <code>put c:\folder\ mybucket -onlydiff -purge -purgeabort:ALL</code> (upload files in <code>c:\folder\</code> to mybucket only if they are different compared to the matching file that is already on S3. Delete files in mybucket that are not in <code>c:\folder\</code> . Do not purge if all S3 files in mybucket would be deleted) <code>put c:\folder\ mybucket -onlydiff -purge -purgeabort:NEVER</code> (upload files in <code>c:\folder\</code> to mybucket only if they are different compared to the matching file that is already on S3. Delete files in mybucket that are not in <code>c:\folder\</code> . Never abort purge)
<code>-move</code>	Move files to S3, e.g. delete local files immediately after they are successfully uploaded to S3. See: How to move files to S3 (difference between -move and -localdelete)	<code>put c:\folder\ mybucket -s -include:*.jpg -move</code> (move all jpg files in <code>c:\folder\</code> and subfolders to mybucket)
<code>-localdelete: COND</code>	Delete local files that: - do not need to be uploaded. - have a corresponding matching file on S3. - for which the condition COND is true. COND is a condition that follows the general condition rules .	<code>put c:\folder\ mybucket -s -onlydiff -localdelete</code> (upload files in <code>c:\folder\</code> and subfolders to mybucket if they are different from files on S3 and delete local files that have a corresponding matching file on S3) <code>put c:\folder\ mybucket -s -onlydiff -localdelete:'age_days > 90'</code> (upload files in <code>c:\folder\</code> and subfolders to mybucket if they are different from

	<p>If the condition COND is not specified, that is, only -localdelete is used, then all local files that have a corresponding matching file on S3 will be deleted.</p> <p>See: How to move files to S3 (difference between -move and -localdelete)</p>	files on S3 and delete local files that have a corresponding matching file on S3 and are older than 90 days)
-include:INCL	Only upload files with path matching the specified mask (Wildcards). Separate multiple masks with " ".	<pre>put c:\folder\ mybucket -include:*.jpg (upload all jpg files in c:\folder\ to mybucket) put c:\folder\ mybucket -include:*.jpg *.gif (upload all jpg and gif files in c:\folder\ to mybucket)</pre>
-exclude:EXCL	Do not upload files with path matching the specified mask (Wildcards). Separate multiple masks with " ".	<pre>put c:\folder\ mybucket -exclude:*.jpg (upload all files in c:\folder\, excluding files with extension .jpg, to mybucket) put c:\folder\ mybucket -exclude:*.jpg *.gif (upload all files in c:\folder\, excluding files with extension .jpg and *.gif, to mybucket)</pre>
-rinclude:INCL	Only upload files with path matching the specified mask (Regular Expression).	<pre>put c:\folder\ mybucket -rinclude:a(x y z)b (upload files in c:\folder\ matching axb, ayb and azb to mybucket) put c:\folder\ mybucket -rinclude:*.gif bmp jpg) (upload files in c:\folder, ending with .gif, .bmp or .jpg, to mybucket) put c:\folder\ mybucket -rinclude:"IMG[P0-9]{4}.jpg" (upload files in c:\folder\ ending with .jpg and starting with IMG and followed by a four-digit number to mybucket)</pre>
-rexclude:EXCL	Do not upload files with path matching the specified mask (Regular Expression).	<pre>put c:\folder\ mybucket -rexclude:[abc] (upload all files in c:\folder\ to mybucket, but exclude files containing a, b or c in the file path)</pre>
-sim	Simulation. Only preview which files would be uploaded, do not actually upload the files yet.	<pre>put c:\folder\ mybucket -include:*.jpg -sim (simulation only, show summary of which files would be selected for upload)</pre>
-showfiles	Show detailed list of all selected files to upload not just the summary.	<pre>put c:\folder\ mybucket -include:*.jpg -sim -showfiles (simulation only, show list of files that would be selected for upload)</pre>
-showdelete	Show detailed list of all selected files to be deleted from the S3 bucket not just the summary. Only applicable if -purge is used.	<pre>put c:\folder\ mybucket -include:*.jpg -purge -sim -showfiles -showdelete (simulation only, show list of files that would be selected for upload and show list of files that would be deleted from the S3 bucket)</pre>
-showlocaldelete	Show detailed list of all selected local files to be deleted from the local folder due to the option -localdelete. Only applicable if -localdelete is used.	<pre>put c:\folder\ mybucket -onlydiff -localdelete:'age_months>6' -showfiles -showlocaldelete -sim (simulation only, show list of files that would be selected for upload from c:\folder\ to mybucket and show list of files that would be deleted from c:\folder\ due to the -localdelete option)</pre>
-showexcl	This flag can only be used in combination with the -sim flag above. Using this flag shows which files would be excluded from the upload.	<pre>put c:\folder\ mybucket -include:*.jpg - sim - showexcl (simulation only, show summary of which files would be selected for upload and list which files would be excluded)</pre>

-noautostatus	Do not automatically show the latest upload status every 10 seconds. The status can be shown by pressing the key 's' while the upload is in process.	put c:\folder\ mybucket -noautostatus (upload files in c:\folder\ to mybucket and do not automatically show the latest upload status every 10 seconds)
-minoutput	Minimal output. Minimize the output that is shown in the S3Express console during a put operation. This option is useful when copying many small files to S3, which could make the S3Express output in the console too fast to read. Minimal output can be toggled on or off at any time during a put operation by pressing the key 'o'.	put c:\folder\ mybucket -minoutput -s
-stoponerror	Stop operation as soon as an error occurs (do not continue with other files).	put c:\folder\ mybucket -s -stoponerror
-optimize	Enable thread optimization for transferring large amounts of relatively small files over fast connections. Recommended to use with at least 4 threads (-t:4).	put c:\folder*.jpg mybucket -s -t:16 -optimize
-accelerate	Use Amazon S3 Transfer Acceleration for this operation. S3Express will use 's3-accelerate.amazonaws.com' as the endpoint for this operation. Transfer Acceleration must to be firstly enabled for the bucket in your account or this option will fail with an error.	put c:\folder*.jpg mybucket -accelerate

 **Notes:**

- **Files** in Windows = **Objects** in S3.

- When uploading files to Amazon S3, the Windows modified timestamp is not kept, because Amazon S3 objects get the time of the upload as modified timestamp. This is part of Amazon S3 functionality and it does not depend on S3Express. In order to keep information re the original file modified timestamp, S3Express adds two custom metadata headers to each uploaded file: `x-amz-meta-s3xpress-modified-time-iso` and `x-amz-meta-s3xpress-modified-time`. The `x-amz-meta-s3xpress-modified-time-iso` header contains the original file timestamp in ISO format, while the `x-amz-meta-s3xpress-modified-time` header contains the original file timestamp in HTTP format. You can see these two metadata headers using the command [getmeta](#) or [ls -showmeta](#).

- If an identical file (i.e. same MD5 value) is already stored on Amazon S3, the file is copied, not uploaded, to save bandwidth. S3Express will show which files were copied (=duplicated) instead of uploaded. This functionality can be disabled using `-nomd5existcheck`

Retry on network error:

The number of retries performed in case of a network error, and the wait time, can be set in the general S3Express options using the command [setopt](#)

13 mkfol (create folder)

mkfol FOLDER

Create S3 folder at [current S3 working location](#).

Parameter	Description	Examples
FOLDER	Name of the folder to be created. A folder on S3 is an empty object that ends with /	mkfol myfolder (create folder <i>myfolder</i> at current working location)

14 lsupl (list multipart uploads)

lsupl [BUCKET_NAME]/[FOLDER]/ [-s]

List in-progress [multipart uploads](#) in a bucket/folder.

Parameter	Description	Examples
[BUCKET_NAME]/[FOLDER]/	Name of the bucket / folder whose in-progress multipart uploads are to be listed. Uploads will be listed with name and upload ID.	lsupl <i>mybucket</i> (list all in-progress multipart uploads in bucket <i>mybucket</i>) lsupl <i>mybucket/subfolder/</i> (list all in-progress multipart uploads in bucket <i>mybucket</i> , subfolder <i>subfolder</i>)
-s	Include all uploads, also in subfolders.	lsupl <i>mybucket -s</i> (list all in-progress multipart uploads in bucket <i>mybucket</i> , including in subfolders)

15 rmupl (remove multipart uploads)

rmupl [BUCKET_NAME]/[FOLDER]/ [-id:UPLOADID] [-file:FILE] [-s]

Remove/abort in-progress [multipart uploads](#) from a bucket.

Parameter	Description	Examples
[BUCKET_NAME]/[FOLDER]/	Name of the bucket / folder whose in-progress multipart uploads are to be removed.	rmupl <i>mybucket</i> (remove all in-progress multipart uploads from bucket <i>mybucket</i>)
-id:UPLOADID	Specify the multipart upload ID to be removed.	rmupl <i>mybucket</i> -id:VXBsb2FkIEIEIGZvciA2aWWpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZA (remove multipart upload with ID VXBsb2FkIEIEIGZvciA2aWWpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZA from bucket <i>mybucket</i>)
-file:FILE	Specify the multipart upload file name to be removed.	rmupl <i>mybucket</i> -file:file.txt (remove multipart upload with name file.txt from bucket <i>mybucket</i>)
-s	Remove uploads recursively in all subfolders too.	rmupl <i>mybucket</i> -s (remove all in-progress multipart uploads from bucket <i>mybucket</i> and from all subfolders of <i>mybucket</i>)

16 del (delete objects)

del [BUCKET_NAME]/[FOLDER]/OBJECT [-s] [-r] [-sim] [-stoponerror] [-cond:"FILTER"] [-include:INCL] [-exclude:EXCL] [-rincl:INCL] [-rexcl:EXCL] [-inclenc] [-exclenc] [-inclrr] [-exclrr] [-inclia] [-exclia] [-inclgl] [-exclgl] [-inclle] [-exclle] [-noconfirm:X] [-version:ID] [-inclversions] [-onlyprev] [-minoutput]

Delete S3 objects.

Parameter	Description	Examples
[BUCKET_NAME]/[FOLDER]/OBJECT	Name of the bucket, folder, object(s) to delete. If not specified, objects in current location are deleted. Wildcard character can be used (i.e. * and ?) like in Windows dir. If object name have spaces, they must be surrounded by quotation marks (") on the command line.	del mybucket/*.txt (delete all objects with extension <i>txt</i> in mybucket) del mybucket/myfolder/*.txt (delete all objects with extension <i>txt</i> in mybucket/myfolder/) del "mybucket/my folder/*.txt" (delete all objects with extension <i>txt</i> in mybucket/my folder/, using quotation marks)
-s	Recursive deleting, e.g. process all subfolders.	del mybucket/* -s (delete all objects from mybucket)
-r	Regular expression. This flag specifies that [BUCKET_NAME]/[FOLDER]/OBJECT must be treated as a regular expression.	del "mybucket/my folder/*.txt .*.vsn" -r (delete all objects with extension <i>txt</i> or <i>vsn</i> in mybucket/my folder/) del mybucket/^r.* -r (delete all objects starting with 'r' in mybucket)
-sim	Only preview which objects would be deleted, do not actually delete the objects.	del mybucket/* -sim (preview object deletion from mybucket)
-stoponerror	Stop deleting files as soon as an error occurs (do not continue with other files).	del mybucket/*.txt -stoponerror (delete all objects with extension <i>txt</i> in mybucket, stop if, and as soon, as an error occurs, e.g. do not continue with other files)
-cond:"FILTER"	Filter condition. Only delete objects matching the specified condition. More info on filter condition syntax and variables.	del mybucket -s -cond:"extract_value(cache-control,'max-age') > 0" (delete all objects in <i>mybucket</i> (recursive, include subfolders) with cache-control:max-age > 0 in the metadata) del mybucket -s -cond:"size = 0" (delete all objects in <i>mybucket</i> (recursive) of size equal to zero) del mybucket -cond:"name starts_with 'a'" (delete all objects in <i>mybucket</i> (non-recursive) with name starting with a) del mybucket -s -cond:"name starts_with 'a' and size > 0" (delete all objects in <i>mybucket</i> (recursive) with name starting with a and size > 0) More info on filter condition syntax and variables
-include:INCL	Only include objects matching specified mask (Wildcards). Separate multiple masks with " ".	del mybucket -include:*.jpg (delete all jpg files in mybucket) del mybucket -include:*.jpg *.gif (delete all jpg and gif files in mybucket)

-exclude: EXCL	Exclude objects matching specified mask (Wildcards). Separate multiple masks with " ".	del mybucket -exclude:*.jpg (delete all files in mybucket but exclude jpg files) del mybucket -exclude:*.jpg *.gif (delete all files in mybucket but exclude jpg files)
-rinclude: INCL	Only include objects matching specified mask (Regular Expression).	del mybucket -rinclude:a(x y z)b (delete files in mybucket matching axb, ayb and azb) del mybucket -rinclude:*.gif *.bmp .jpg (delete files in mybucket matching anything ending with .gif, .bmp or .jpg) del mybucket -rinclude:"IMGP[0-9]{4}.jpg" (delete files in mybucket ending with .jpg, starting with IMG and followed by a four-digit number)
-rexclude: EXCL	Exclude objects matching specified mask (Regular Expression).	del mybucket -rexclude:[abc] (delete all files in mybucket but exclude files containing a, b or c)
-inclenc -exclenc	Include only server-side encrypted files. Exclude server-side encrypted files.	del mybucket -inclenc (delete all files in mybucket that are server-side encrypted) del mybucket -exclenc (delete all files in mybucket that are NOT server-side encrypted)
-inclrr -exclrr	Include only reduced redundancy files. Exclude reduced redundancy files.	del mybucket -inclrr (delete all files in mybucket that are reduced redundancy) del mybucket -exclrr (delete all files in mybucket that are NOT reduced redundancy)
-inclia -exclia	Include only infrequent access files. Exclude infrequent access files.	del mybucket -inclia (delete all files in mybucket that are infrequent access) del mybucket -exclia (delete all files in mybucket that are NOT infrequent access)
-inclgl -exclgl	Include only Glacier files. Exclude Glacier files.	del mybucket -inclgl (delete all files in mybucket that are part of Amazon Glacier) del mybucket -exclgl (delete all files in mybucket that are NOT part of Amazon Glacier)
-inclle -exclle	Include only client-side (locally) encrypted files. Exclude only client-side (locally) encrypted files.	del mybucket -inclle (delete all files in mybucket that were locally encrypted) del mybucket -exclle (delete all files in mybucket that were NOT locally encrypted)
-noconfirm:X	Use <i>-noconfirm</i> to disable the deletion confirmation request "Confirm deletion of ..." that appears if more than 1 object is selected to be deleted. You can use also <i>-noconfirm</i> with a value, e.g. <i>-noconfirm:10</i> . This would disable the confirmation request only for up to 10 files: if more than 10 files are selected to be deleted the confirmation question is still shown.	del mybucket -s -noconfirm (delete all files in mybucket and subfolders and do not ask for confirmation) del mybucket -s -noconfirm:100 (delete all files in mybucket and subfolders and do not ask for confirmation if 100 or less files are selected to be deleted. Ask for confirmation if more than 100 files are selected to be deleted.)
-inclversions	Include object previous versions (for buckets with object versioning enabled).	del mybucket/*.txt -inclversions (delete all objects with extension txt in mybucket and also all previous versions of the objects)
-version:ID	Specify version ID of object to be deleted (for buckets with object versioning enabled).	del mybucket/file.txt -version:23443232 (delete object file.txt, object version ID 23443232, in mybucket)

-onlyprev	Include only the previous versions of an object. All current versions of objects are excluded (for buckets with object versioning enabled).	del mybucket/*.txt -onlyprev -cond:"s3_age_months>6" (delete all and only previous versions of objects with extension txt in mybucket, which are older than 6 months)
-minoutput	Minimal output. Minimize the output that is shown in the S3Express console during a delete operation. Only the total files deleted and eventual errors will be shown.	del mybucket/* -s -minoutput (delete all objects from mybucket, show minimal output)

**Notes:**

- Use quotation marks (") if folder or object names contain blank spaces, e.g. del "mybucket/my folder/name with a space.txt"
- If multiple files are to be deleted, file deletion will be done using multiple concurrent threads. The maximum threads to use can be specified with the command [setopt](#) , option -qmaxthreads

17 copy (copy object)

```
copy [BUCKET]/[FOLDER]/FROMOBJECT [BUCKET]/[FOLDER]/
TOOBJECT [-cacl:CANNED_ACL] [-meta:METADATA] [-e] [-rr] [-ia] [-
keep:KEEP]
```

Make a copy of one S3 object and optionally apply new ACL and/or new metadata or keep existing.
Note that the original object is not removed (if needed it must be removed with the [del](#) command).

Parameter	Description	Examples
-cacl: CANNED_ACL	<p>Apply canned ACL to copy of object. Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions.</p> <p>Valid Values for CANNED_ACL:</p> <p>private (Owner gets FULL CONTROL. No one else has access rights, this is the default for an object)</p> <p>public-read (Owner gets FULL CONTROL. The AllUsers group, that is everyone, gets READ access)</p> <p>public-read-write (Owner gets FULL CONTROL. The AllUsers group, that is everyone, gets READ and WRITE access)</p> <p>authenticated-read (Owner gets FULL CONTROL. The AuthenticatedUsers group, that is all Amazon AWS accounts, gets READ access.)</p> <p>bucket-owner-read (Object owner gets FULL CONTROL. Bucket owner gets READ access)</p> <p>bucket-owner-full-control (Both the object owner and the bucket owner get FULL CONTROL over the object)</p> <p>Note: You can specify only one of these canned ACLs in your request.</p>	<p>copy mybucket/myfile.txt mybucket/myfilecopy.txt (duplicate file mybucket/myfile.txt to mybucket/myfilecopy.txt)</p> <p>copy mybucket/myfile.txt mybucket/myfilecopy.txt -cacl:public-read (duplicate file mybucket/myfile.txt to mybucket/myfilecopy.txt and apply 'public-read' ACL to target mybucket/myfilecopy.txt)</p>
-meta:META	<p>Metadata headers to be added to copy of object. Multiple</p>	<p>copy mybucket/myfile.txt mybucket/myfilecopy.txt -meta:"cache-control:max-age=60" (duplicate file</p>

	metadata headers should be separated by .	mybucket/myfile.txt to mybucket/myfilecopy.txt and apply metadata header "cache-control:max-age=60" to target mybucket/myfilecopy.txt)
-e	Apply Amazon S3 Server Side Encryption to copy of object.	copy mybucket/myfile.txt mybucket/subfolder/myfile.txt -e (duplicate file mybucket/myfile.txt to mybucket/subfolder/myfile.txt and apply server side encryption to target mybucket/subfolder/myfile.txt)
-rr	Set S3 storage class to "Reduced Redundancy" for copy of object.	copy mybucket/myfile.txt mybucket/subfolder/myfile.txt -rr (duplicate file mybucket/myfile.txt to mybucket/subfolder/myfile.txt and apply storage class Reduced Redundancy to target mybucket/subfolder/myfile.txt)
-ia	Set S3 storage class to "Infrequent Access" for copy of object.	copy mybucket/myfile.txt mybucket/subfolder/myfile.txt --ia (duplicate file mybucket/myfile.txt to mybucket/subfolder/myfile.txt and apply storage class Infrequent Access to target mybucket/subfolder/myfile.txt)
-keep:KEEP	Copy also metadata and/or ACL from source object to copy of object. Use <i>-keep:acl</i> to copy the existing ACL Use <i>-keep:meta</i> to copy the existing metadata Use <i>-keep</i> to copy both, metadata and ACL. If the <i>-keep</i> parm is not specified, metadata and ACL are not copied.	copy mybucket/myfile.txt mybucket/subfolder/myfile.txt -keep (duplicate file mybucket/myfile.txt to mybucket/subfolder/myfile.txt and copy also metadata and ACL from source mybucket/myfile.txt to target mybucket/subfolder/myfile.txt)

18 restore (restore objects)

```
restore [BUCKET_NAME]/[FOLDER]/OBJECT -days:X [-tier:X] [-s] [-r]
[-sim] [-stoponerror] [-noconfirm:X] [-cond:"FILTER"] [-include:
INCL] [-exclude:EXCL] [-rincl:INCL] [-rexclude:EXCL] [-inclenc] [-
exclenc] [-inclle] [-exclle] [-inclgl] [-exclgl] [-version:ID] [-
inclversions] [-onlyprev]
```

Restore copies of archived objects. Specify the number of days you want the object copy restored for.

Parameter	Description	Examples
[BUCKET_NAME]/ [FOLDER]/ OBJECT	Name of the bucket, folder, object(s) to restore. If not specified, objects in current location are restored. Wildcard character can be used (i.e. * and ?) like in Windows dir. If object name have spaces, they must be surrounded by quotation marks (") on the command line.	restore mybucket/*.txt -days:1 (restore all objects with extension <i>txt</i> in mybucket for 1 day) restore mybucket/myfolder/*.txt -days:2 (restore all objects with extension <i>txt</i> in mybucket/myfolder/ for 2 days) restore "mybucket/my folder/*.txt" -days:2 (restore all objects with extension <i>txt</i> in mybucket/my folder/ for 2 days, using quotation marks)
-days:X	Required. The number of days you want the object copy restored.	restore mybucket/myfile.txt -days:10 (restore myfile.txt in mybucket for 10 days)
-tier:X	Optional. When restoring archived objects, you can specify one of the following data access tier options with the -Tier parameter: Expedited Standard Bulk. The default value, if not specified, is Standard. See the Amazon S3 documentation for more information.	restore mybucket/a.txt -days:5 -tier:Expedited restore *.jpg -s -days:10 -tier:Bulk
-s	Recursive restore, e.g. process objects in all subfolders.	restore mybucket/* -s -days:1 (restore all objects in mybucket and subfolders for 1 day)
-r	Regular expression. This flag specifies that [BUCKET_NAME]/[FOLDER]/OBJECT is to be treated as a regular expression.	restore "mybucket/my folder/**.txt **.vsn" -r -days:2 (restore all objects with extension <i>txt</i> or <i>vsn</i> in mybucket/my folder/ for 2 days) restore mybucket/^r.* -r -days:2 (restore all objects starting with 'r' in mybucket for 2 days)
-sim	Only preview which objects would be restored, do not actually restore the objects.	restore mybucket/* -sim -days:10 (preview object restore of all files in mybucket)
-stoponerror	Stop restoring files as soon as an error occurs (do not continue with other files).	restore mybucket/*.txt -stoponerror -days:1 (restore all objects with extension <i>txt</i> in mybucket for 1 day, stop if and as soon as an error occurs, e.g. do not continue with other files)
-noconfirm:X	Use <i>-noconfirm</i> to disable the	restore mybucket -s -noconfirm -days:7 (restore all files

	restore confirmation request "Confirm restore of ..." that appears if more than 1 object is selected to be restored. You can use also <code>-noconfirm</code> with a value, e.g. <code>-noconfirm:10</code> . This would disable the confirmation request only for up to 10 files: if more than 10 files are selected to be restored the confirmation question is still shown.	in mybucket and subfolders for 7 days and do not ask for confirmation) restore mybucket -s -noconfirm:100 -days:3 (restore all files in mybucket and subfolders for 3 days and do not ask for confirmation if 100 or less files are selected to be restored. Ask for confirmation if more than 100 files are selected to be restored.)
-cond: "FILTER"	Filter condition. Only restore objects matching the specified condition. More info on filter condition syntax and variables.	restore mybucket -s -cond:"extract_value(cache-control,'max-age') > 0" -days:1 (restore all objects in <i>mybucket</i> (recursive, include subfolders) for 1 day if cache-control:max-age > 0 in the metadata) restore mybucket -s -cond:"size = 0" -days:2 (restore all objects in <i>mybucket</i> (recursive) if size equal to zero) restore mybucket -cond:"name starts_with 'a'" -days:7 (restore all objects in <i>mybucket</i> (non-recursive) if name starting with a) restore mybucket -s -cond:"name starts_with 'a' and size > 0" -days:10 (restore all objects in <i>mybucket</i> (recursive) with name starting with a and size > 0) More info on filter condition syntax and variables
-include:INCL	Only include objects matching specified mask (Wildcards). Separate multiple masks with " ".	restore mybucket -include:*.jpg -days:10 (restore all jpg files in mybucket for 10 days) restore mybucket -include:*.jpg *.gif -days:1 (restore all jpg and gif files in mybucket for 1 day)
-exclude: EXCL	Exclude objects matching specified mask (Wildcards). Separate multiple masks with " ".	restore mybucket -exclude:*.jpg -days:1 (restore all files in mybucket but exclude jpg files) restore mybucket -exclude:*.jpg *.gif -days:1 (restore all files in mybucket but exclude jpg files)
-rinclude: INCL	Only include objects matching specified mask (Regular Expression).	restore mybucket -rinclude:a(x y z)b -days:1 (restore files in mybucket matching axb, ayb and azb) restore mybucket -rinclude:*(.gif bmp jpg) -days:1 (restore files in mybucket matching anything ending with .gif, .bmp or .jpg) restore mybucket -rinclude:"IMG[P0-9]{4}.jpg" -days:1 (restore files in mybucket ending with .jpg, starting with IMG and followed by a four-digit number)
-rexclude: EXCL	Exclude objects matching specified mask (Regular Expression).	restore mybucket -rexclude:[abc] -days:1 (restore all files in mybucket but exclude files containing a, b or c)
-inclenc -exclenc	Include only server-side encrypted files. Exclude server-side encrypted files.	restore mybucket -inclenc -days:1 (restore all files in mybucket that are server-side encrypted) restore mybucket -exclenc -days:1 (restore all files in mybucket that are NOT server-side encrypted)
-inclgl -exclgl	Include only Glacier files. Exclude Glacier files.	restore mybucket -inclgl -days:1 (restore all files in mybucket that are part of Amazon Glacier)
-inclle -exclle	Include only client-side (locally) encrypted files. Exclude only client-side (locally) encrypted files.	restore mybucket -inclle -days:1 (restore all files in mybucket that were locally encrypted) restore mybucket -exclle -days:1 (restore all files in mybucket that were NOT locally encrypted)
-version:ID	Specify version ID of object to be restored (for buckets with object	restore mybucket/file.txt -version:23443232 -days:2 (restore object file.txt, object version ID 23443232, in

	versioning enabled).	mybucket, for 2 days)
-inclversions	Include object previous versions (for buckets with object versioning enabled).	restore mybucket/*.txt -inclversions -days:9 (restore all objects with extension txt in mybucket and also all previous versions of the objects for 9 days)
-onlyprev	Include only the previous versions of an object. All current versions of objects are excluded (for buckets with object versioning enabled).	restore mybucket/*.txt -onlyprev -days:5 -cond:"s3_age_months<=6" (restore all and only previous versions of objects with extension txt in mybucket, which are dated less than 6 months ago, for 5 days)

**Notes:**

- Use quotation marks (") if folder or object names contain blank spaces, e.g. restore "mybucket/my folder/name with a space.txt"
- If multiple files are to be restored, the operation will be performed using multiple concurrent threads. The maximum threads to use can be specified with the command [setopt](#), option `-qmaxthreads`

19 Authorization Commands

The following commands are used to set, save, load, delete Amazon S3 authorizations, that is, Access Key ID and Secret Access Key pairs.

saveauth ACCESS_KEY_ID SECRET_ACCESS_KEY [NAME]

Save Access Key ID and Secret Access Key in S3Express.

Access Key ID and Secret Access Key are stored encrypted in the Windows Registry and can then be recalled with the command **loadauth**.

Parameter	Description
ACCESS_KEY_ID	Required. This is the Amazon Access Key ID that S3Express should use to connect to Amazon S3.
SECRET_ACCESS_KEY	Required. This is the corresponding Amazon Secret Access Key that S3Express should use to connect to Amazon S3.
NAME	Optional. The name for this authorization. A name can be used to store multiple authorizations in S3Express. If a name is not use the ACCESS_KEY_ID and SECRET_ACCESS_KEY are saved without a name, as default authorization.

loadauth [NAME]

Load a previously saved Access Key ID and Secret Access Key in S3Express for use.

Parameter	Description
NAME	Optional. The name of the authorization to load. If a name is not specified the default ACCESS_KEY_ID and SECRET_ACCESS_KEY are loaded.

showauth [NAME]

Show previously saved Access Key ID and Secret Access Key in S3Express.

Parameter	Description
NAME	Optional. The name of the authorization to show. If a name is not specified the default ACCESS_KEY_ID and SECRET_ACCESS_KEY are shown. Note: You can show all authorizations saved in S3Express using command: showauth <all>

rmauth [NAME]

Remove previously saved Access Key ID and Secret Access Key from S3Express.

Parameter	Description
NAME	Optional. The name of the authorization to remove. If a name is not specified the default ACCESS_KEY_ID and SECRET_ACCESS_KEY are removed. Note: You can remove all authorizations saved in S3Express using command:

```
rmauth <all>
```

Authorization Examples:

To save the Access Key ID and Secret Access Key pair *FASWQDSDSSSZXAS1SA* and *AsFZEDy2BQfFSFzFfgKyyOF/xCaRcK4RMc* as the default S3Express authorization use command:
saveauth FASWQDSDSSSZXAS1SA AsFZEDy2BQfFSFzFfgKyyOF/xCaRcK4RMc

This authorization can then be recalled each time by using command:
loadauth

Note that by default S3Express loads the latest used authorization at startup automatically, so once you saved the pair *FASWQDSDSSSZXAS1SA* and *AsFZEDy2BQfFSFzFfgKyyOF/xCaRcK4RMc* using *saveauth FASWQDSDSSSZXAS1SA AsFZEDy2BQfFSFzFfgKyyOF/xCaRcK4RMc*, S3Express will automatically reload it every time it starts, no need to use *loadauth* each time.

To remove this authorization from S3Express, use:
rmauth

To show this authorization in S3Express, use:
showauth

You can also save multiple authorizations in S3Express. For instance you could have:
saveauth FASWQDSDSSSZXAS1SA AsFZEDy2BQfFSFzFfgKyyOF/xCaRcK4RMc
but also:
saveauth 1FASWQDSDSSSZXAS1SA 1AsFZEDy2BQfFSFzFfgKyyOF/xCaRcK4RMc MYAUTH1
and
saveauth 2FASWQDSDSSSZXAS1SA 2AsFZEDy2BQfFSFzFfgKyyOF/xCaRcK4RMc MYAUTH2

You would then load the required authorization using the *loadauth* command ,e.g.:
loadauth
or
loadauth MYAUTH1
or
loadauth MYAUTH2

Note that by default S3Express loads the latest used authorization at startup automatically, so after loading *MYAUTH2*, that authorization would be reloaded automatically at next S3Express startup.

To remove the *MYAUTH1* authorization from S3Express, use:
rmauth MYAUTH1

To show the *MYAUTH1* authorization, use:
showauth MYAUTH1

To show all authorizations saved in S3Express, use:
showauth <all>

To remove all authorizations saved in S3Express, use:
rmauth <all>

20 setopt / showopt

The following commands are used to set (or reset) S3Express options.

```
setopt [-retry:X] [-retrywait:X] [-verbosity:X] [-timer:on|off] [-
endpoint:ENDPOINT] [-region:REGION] [-protocol:PROTOCOL] [-
clientencpwd:PASSWORD] [-clientencpwdhint:HINT] [-
clientencprogram:PROGRAM] [-proxyserver:SERVER] [-qmaxthreads:
X] [-timeout:X] [-useV4sign:on|off] [-disablecertvalidation:on|off] [-
usepathstyle:on|off] [-reset]
```

Set S3Express options. Options are saved in the Windows registry and are re-used every time S3Express starts, unless changed again.

Option	Description	Default Value
-retry:X	Set the number of retries performed by S3Express in the case of a network error. By default S3Express retries 3 times.	3
-retrywait:X	Set the wait time, in seconds, before a retry. Default value is 5 seconds.	5
-verbosity:X	Set the output verbosity level. Default value is 1. It can be increased to 2 or 3.	1
-timer:on off	Set the timer on or off (use -timer:on to set the timer on, -timer:off to set the timer off). When the timer is on, the elapsed time for each command is included in the output. By default the timer is off.	off
-endpoint:ENDPOINT	Set the service endpoint that S3Express will use for all requests. The default endpoint is <i>s3.amazonaws.com</i>	s3.amazonaws.com
-region:REGION	Set the service region. Optional.	
-protocol:PROTOCOL	Set the protocol S3Express should use to connect to Amazon S3 servers. Possible values are <i>https</i> (the default value) or <i>http</i> .	https
Encryption Options		
-clientencpwd:PASSWORD	Password to use for local file encryption. See -le flag of the put command.	not set
-clientencpwdhint:HINT	An optional password hint to use during local file encryption. See -le flag of the put command for more info on local file encryption. A password hint is a reminder of how you made up your encryption password and it's added to the metadata of each encrypted file. The metadata header containing the password hint is 'x-amz-meta-s3xpress-encrypted-pwd-hint'. If you forgot the encryption password used to encrypt a file, you can try to use the hint to remember it. The password hint can not contain the password itself, because it is attached to the file metadata unencrypted.	not set

<p>-clientencprogram: PROGRAM</p>	<p>Use this option to specify which program S3Express should use to perform local file encryption (and optionally, depending on the program capabilities, file compression).</p> <p>The default value and the default encryption program used by S3Express is <i>aescrypt</i>. S3Express will use the program <i>aescrypt.exe</i> to perform local file encryption. AEScrypt is an open-source encryption program that can be downloaded from www.aescrypt.com. Download the command line version of AEScrypt for Windows and save the file <i>aescrypt.exe</i> in the same folder where <i>S3Express.exe</i> is located.</p> <p>Alternatively, you can set the <i>-clientencprogram</i> option to <i>7zip</i>. In this case, S3Express will use the program <i>7z.exe</i> to encrypt and it also compresses files locally before uploading. <i>7zip</i> is also open source and can be downloaded from here: www.7-zip.org. Once installed, copy the file <i>7z.exe</i> to the folder where <i>s3express.exe</i> is located and set the <i>-clientencprogram</i> option to <i>7zip</i>, e.g.:</p> <pre>setopt -clientencprogram:7zip</pre> <p>Finally, you can specify your own custom program to use in the following form: <code><program path> <program parameters></code>, e.g. <code>setopt -clientencprogram:<program_path> <program_parameters></code> <code><program_path></code> is the path of the encryption program to use. <code><program_parameters></code> are the parameters to be passed to the encryption program. They are to be separated by a <code> </code>. <code><program_parameters></code> should contain at least 3 tags: <code><password> <output> <input></code></p> <p>For instance, for <i>aescrypt</i> <code><program_path></code> and <code><program_parameters></code> would look like this: <pre>setopt -clientencprogram:"aescrypt.exe -e -p <password> -o <output> <input>"</pre> <i>(aescrypt.exe is in the same folder as s3express.exe, so the program path is just aescrypt.exe)</i> For <i>7zip</i>, <code><program_path></code> and <code><program_parameters></code> would look like this: <pre>setopt -clientencprogram:"7z.exe a -y -p<password> <output> <input>"</pre></p>	<p>aescrypt</p>
<p>Advanced Options</p>		
<p>-proxyserver:SERVER</p>	<p>Set the proxy server S3Express should use to connect to the Internet.</p> <p>If this value is set to <i>auto</i> (the default value), the proxy server address is automatically taken from the Windows Internet settings. S3Express will look for the system proxy settings for <i>https</i> connections. If not present in the system settings, no proxy server will be used.</p> <p>If this value is set to <i>direct</i>, no proxy server is used.</p> <p>Alternatively you can manually specify a proxy server for S3Express to use. This can be in one of the following</p>	<p>auto</p>

	<p>forms:</p> <p><i>server</i> (just the proxy server address, e.g. proxyserver.com or 10.1.1.24)</p> <p><i>server:port</i> (the proxy server address and port, e.g. proxyserver.com:8000 or 10.1.1.24:3128)</p> <p><i>username:password@server:port</i> (the proxy server address and port, plus username and password, e.g. marc:mypassword@proxyserver.com:3128)</p> <p>See some examples below.</p>	
-qmaxthreads:X	Sets the maximum amount of concurrent threads to use when querying multiple S3 objects or when deleting multiple S3 objects. By default this value is set to 20. This value does not affect the maximum amount of concurrent threads to use during file uploads (i.e. the <i>put</i> command). To control the maximum amount of concurrent threads to use during file uploads, use the <i>-t</i> parameter of the <i>put</i> command.	20
-timeout:X	Set the timeout (in seconds) of each communication between S3Express and Amazon S3. The default value is 60 seconds. Set timeout to 0 to disable timeout (not recommended). If no data is exchanged between S3Express and Amazon S3 within the time specified by the <i>-timeout</i> option, then the request is aborted. A new request is initiated if the <i>-retry</i> option allows it (see above).	60
-useV4sign:on off	Use the latest Amazon S3 signature version 4. If using an old Amazon-S3 compatible service which does not support signature version 4, set this option to off.	on
-disablecertvalidation:on off	Disable SSL certificate validation over https.	off
-usepathstyle:on off	Use path-style requests instead of virtual hosted-style requests. This option might be needed when working with S3 compatible services that do not support virtual hosted-style requests.	off
Reset Options		
-reset	Reset option values to default. To reset a specific option only, specify the option on the command line, e.g. <i>setopt -verbosity -reset</i> or <i>setopt -proxyserver -reset</i> . If no option is specified, e.g. <i>setopt -reset</i> , then all options are reset to default values.	n.a.

Note: Once set, options are saved in the Windows Registry and are re-used every time S3Express starts, unless changed again.

The encryption password is stored in the Windows Registry encrypted.

showopt

Show the current value for specific S3Express options or for all options.

To show specific options only, specify the options on the command line, e.g. *showopt -verbosity* or *setopt -*

proxyserver or both together showopt -verbosity -proxyserver.
If no option is specified, e.g. using just showopt, then all values of all options are shown. Values that are not the default values are highlighted.

setopt examples:

```
setopt -retry:10 -retrywait:60
setopt -clientencpwd:mypassword
setopt -reset
setopt -verbosity -reset
setopt -timer:on
setopt -endpoint:s3-website-us-gov-west-1.amazonaws.com
setopt -proxyserver:marc:mypassword@proxyserver.com:3128
setopt -proxyserver:10.0.1.254:443
setopt -proxyserver:auto
setopt -proxyserver:direct
setopt -protocol:http
setopt -protocol:https
setopt -clientencprogram:7zip
setopt -disablecertvalidation:on
```

showopt examples:

```
showopt
showopt -verbosity
setopt -proxyserver
showopt -verbosity -proxyserver
```

21 license (enter license)

The license command is used to enter a license in S3Express. Entering a license unlocks the S3Express trial.

```
license "LICENSE_TEXT" "LICENSE_KEY"
```

Option	Description
LICENSE_TEXT	Required. This is the license text as provided via e-mail. Surround with double quotation marks. Do not use [].
LICENSE_KEY	Required. This is the license key as provided via e-mail. Surround with double quotation marks. Do not use [].

Examples:

```
license "Leoclara-Inc-USA" "213-12-111A8"
```

22 exec (execute commands from file)

The exec command is used to load and execute a list of commands from a text file.

exec FILE_NAME [-tofile:OUTPUT_FILENAME]

Option	Description
FILE_NAME	A file name is required. This is the name of a text file, saved with standard ASCII or UTF-8 encoding, that contains a list of S3Express commands to be executed. Each command must be on a separate line. If there are comment lines in the file, they must start with character #. If the file name contains spaces, then it must be surrounded by double quotes (").
[-tofile:OUTPUT_FILENAME]	Optionally log output to a file. See Redirect Output to a File

Examples

exec *commands.txt* (load and execute commands from file "commands.txt" in the current directory, usually the same directory where S3Express.exe is, unless it was changed)

exec "c:\folder\subfolder A\my commands.txt" (load and execute commands from file "c:\folder\subfolder A\my commands.txt")

exec "c:\folder\subfolder A\my commands.txt" -tofile:"c:\folder\subfolder A\output.txt" (load and execute commands from file "c:\folder\subfolder A\my commands.txt". Redirect S3Express output to "c:\folder\subfolder A\output.txt")

A) Example of a Text File Containing Commands

```
# Comment <- this is a comment
ls
put c:\folder\ mybucket -onlydiff -keep
quit
# Another comment
```

B) Example of a Text File Containing Commands

```
# Set OnErrorSkip to ON. See OnErrorSkip for more details.
OnErrorSkip ON

# Put first folder. If error it will skip the other commands, because OnErrorSkip ON was set
put c:\folderA\ mybucket\folderA\ -s -onlydiff -purge

# Pause 10 seconds. See Pause command for more details.
pause 10

# Put second folder.
put c:\folderB\ mybucket\folderB\ -s -onlydiff -purge

# finished
quit
```

C) Example of a Text File Containing Commands

```
# Put first folder. If error, it will NOT skip the other commands, because OnErrorSkip ON was not set.
put c:\folderA\ mybucket\folderA\ -s -onlydiff -purge

# Pause 10 seconds. See Pause command for more details.
pause 10

# Put second folder.
put c:\folderB\ mybucket\folderB\ -s -onlydiff -purge

# Whatever happened, I do not want the exit code to report an error, use ResetErrorStatus
ResetErrorStatus

# finished
quit
```

Notes:

- If one of the commands in the text file fails (i.e. it returns an error), the other commands are still executed, unless you specify *OnErrorSkip ON*.
- *S3Express.exe* returns 0 if all executed commands were successful. It will return 1 otherwise. You can reset the error status with command *ResetErrorStatus*.

23 Other Commands

checkupdates

Check for program updates. This command opens a web browser and shows if there are more up-to-date versions of S3Express available for download.

Example:

```
checkupdates
```

md5 FILE_NAME [-mul:PARTSIZE]

Calculate and show MD5 value of a local file.

Option	Description
FILE_NAME	Required. Local file name whose MD5 value must be calculated.
-mul:PARTSIZE	Optional flag. Calculate MD5 in multi-part mode. The PARTSIZE value is optional and can be used to specify the size of each upload part to use, in Megabytes. The minimum upload part size is 5MB and that is also the default size used if PARTSIZE is not specified. Max size is 1000 Megabytes. This is the same flag as used in the <i>put</i> command.

Example:

```
md5 c:\folderA\test.txt
md5 "c:\folderA\name with spaces.txt"
```

mimetype EXTENSION

Show the default MIME type used by S3Express for a specific file extension.

Option	Description
EXTENSION	Required. File extension.

Example:

```
mimetype .exe
mimetype .html
mimetype .jpg
mimetype .css
```

OnErrorSkip ON/OFF

Set S3Express error handling behavior when processing multiple commands (see [exec](#) and [scripting via command line](#))

Option	Description
ON/OFF	OnErrorSkip ON (if one command is unsuccessful, skip all other commands) OnErrorSkip OFF (this is the default. If one command is unsuccessful, continue with all other commands)

Example:

```
OnErrorSkip ON
OnErrorSkip OFF
```

ResetErrorStatus

Reset error status to success when processing multiple commands (see [exec](#) and [scripting via command line](#))

Example:

```
ResetErrorStatus
```

ShowErrorStatus

Show current error status when processing multiple commands (see [exec](#) and [scripting via command line](#))

Example:

```
ShowErrorStatus
```

pause SECONDS

Pause for the specified amount of seconds when processing multiple commands (see [exec](#) and [scripting via command line](#))

Option	Description
SECONDS	SECONDS specifies the amount of seconds to pause.

Example:

```
pause 10 (pause for 10 seconds)
pause 1 (pause for 1 second)
```


pwd

Show current local working directory.

Example:

```
pwd
```

24 Filter Condition Syntax (-cond:"FILTER")

Filter conditions are expressions that can evaluate to *true* or *false*. They are used to filter objects for commands that support the flag **-cond:"FILTER"**, such as [ls](#), [setacl](#), [setmeta](#), [put](#), [del](#).

OPERATORS

Operators	Explanation	Example
=	Equal	size = 10 (true if file's size is 10 bytes)
<> , !=	Not equal	year != 2013 (true if file's year is not equal to 2013) month <> 1 (true if file's month is not equal to 1)
> , >=, <, <=	Greater than, greater than or equal, smaller than, smaller than or equal	sizemb >= 20 AND sizemb <= 30 (true if file's size in MB is between 20 and 30)
+ , - , * , /	Addition, subtraction, multiplication, division	hour + 10 < 20 (true if file's hour plus 10 equal 20)
%	Modulo operation: finds the remainder of division of one number by another	curr_day % 2 = 0 (true on even days) curr_day % 2 = 1 (true on odd days) curr_day % 5 = 0 (true every 5 days)
OR , AND	Logical operators. Group parenthesis for precedence.	sizemb > 5 AND (year < 2010 OR year > 2013)
isoneof	True if a value is contained in a list of values. The list of values must be enclosed in { } and each value separated by ; or , For text values, * can be used a wildcard to match any text.	day isoneof {1,2,3,4,5,6,7,8} (true if file's day is 1,2,3,4,5,6,7 or 8) month isoneof {1;3;7;12} (true if file's month is 1,3,7 or 12) name isoneof {'*.css','*.ini'} (true if file's name matches *.css or *.ini)
isnotoneof	True if a value is NOT contained in a list of values. The list of values must be enclosed in { } and each value separated by ; or ,	day isnotoneof {1,2,3,4,5,6,7,8} (true if day is not 1,2,3,4,5,6,7 or 8) name isnotoneof {'*.css','*.ini'} (true if file's name does not match *.css or *.ini)
contains	True if a text value contains another text value	name contains 'ab' (true if file's name contains 'ab'. Note that comparison is case insensitive)
starts_with	True if a text value starts with another text value	s3_name starts_with 'A' (true if S3 file's name starts with 'A' or 'a', comparison is case insensitive)
ends_with	True if a text value ends with another text value	s3_name ends_with 'Z' (true if S3 file's name ends with 'Z' or 'z', comparison is case insensitive)

matches	True if a text value matches another text value. Using * matches any text at that location	name matches '*.txt' OR name matches '*.jpg'
regex_matches	True if a text value matches another text value using regular expression syntax.	name regex_matches '^A(.*) ^B(.*)' (true if file's name starts with 'A' or 'B', match is case sensitive, using regular expressions) name regex_matches '(?i)^A(.*) ^B(.*)' (true if file's name starts with 'A', 'B', 'a' or 'b'. (?i) makes the match case insensitive)

LOCAL FILE VARIABLES (ONLY USABLE IN THE CONDITION FOR THE PUT COMMAND)

Local File Variable	Explanation	Example
day	The file's day of the month, in the range 1 through 31	put c:\folder\ mybucket -cond:"day = 15 OR dayofweek = 1" (upload all files in c:\folder\ that were lastly modified on the 15th day of the month or on the first day of the week)
month	The file's month of the year, in the range 1 through 12 (1 = January, 12 = December)	put c:\folder\ mybucket -cond:"month = 1" (upload all files in c:\folder\ that were lastly modified date in January)
year	The file's year, 1970 to 2038	put c:\folder\ mybucket -cond:"year <> 2012" (upload all files in c:\folder\ that were not lastly modified in 2012)
date	The file's date: 'YYYY-MM-DD'	date = '2012-04-03' (true if file's date is '2012-04-03') date matches '2012-04-*' (true if file's date is in April 2012) Note that <i>date</i> is a text value and values must be surrounded by two '
time	The file's time: 'HH:MM:SS'	time matches '08:12:10' (true if file's time matches '08:12:10')
hour	The file's hour, in the range 0 through 23	put c:\folder\ mybucket -cond:"hour >= 0 AND hour <= 7" (upload all files in c:\folder\ that were lastly modified between midnight and 7:59 am)
minute	The file's minute, in the range 0 through 59	put c:\folder*.jpg mybucket -cond:"minute >= 30" (upload all JPG files in c:\folder\ that were lastly modified in the second half of each hour)
second	The file's second, in the range 0 through 59	put c:\folder*.jpg mybucket -cond:"second >= 30" (upload all JPG files in c:\folder\ that were lastly modified in the second half of each minute)

dayofweek	The file's day of the week (0 = Sunday, 1 = Monday, etc. to 6 = Saturday)	dayofweek = 0 (true if file's modified timestamp is on a Sunday). dayofweek = 0 OR dayofweek = 6 (true for Sundays and Saturdays). dayofweek <> 1 (true if file's day of week is not Monday).
dayofyear	The file's day of the year. Number from 1 to 366	dayofyear = 32 (true if file's modified timestamp is on 1st of February) dayofyear % 2 = 0 (true file's modified timestamp is on even days)
weeknumber	The file's week number (ISO 8601)	weeknumber = 32 (true if file's timestamp is on the 32nd week of the year)
endofmonth	The file's end of month's date: 'YYYY-MM-DD'	endofmonth = "2012-04-30" (true if file's timestamp is in April 2012) endofmonth matches "*-*-30" (true if file's timestamp is in a month with 30 days)
age_months	The file's age in months.	put c:\folder*.jpg mybucket - cond:"age_months > 6" (upload all JPG files in c:\folder\ that were lastly modified more than 6 months ago)
age_days	The file's age in days.	put c:\folder*.gif mybucket -cond:"age_days < 21" (upload all GIF files in c:\folder\ that were lastly modified less than 21 days ago)
age_hours	The file's age in hours.	age_hours > 1 (true if file is more than 1 hour old)
age_mins	The file's age in minutes.	age_mins <= 30 (true if file is less than 30 minutes old)
age_secs	The file's age in seconds.	age_secs >= 2000 (true if file is more than 2000 seconds old)
name	Local file name	put c:\folder\ mybucket -cond:"name starts_with 'a'" (upload all files in c:\folder\ whose name starts with 'a')
path	Local file path (includes folder)	put c:\folder\ mybucket -cond:"path contains '\subfolder\'" (upload all files in c:\folder\ whose path contains '\subfolder\'', e.g. in the subfolder 'subfolder')
extension	Local file extension	put c:\folder\ mybucket -cond:"extension <> '.db'" (upload all files in c:\folder\ with file extension different from '.db')
stem	Local file stem (file name without extension)	put c:\folder\ mybucket -cond:"stem <> 'Read'" (upload all files in c:\folder\ with file

		stem different from 'read'. Note that <> is case sensitive)
size	File size in bytes	put c:\folder\ mybucket -cond:"size > 20000" (upload all files in c:\folder\ whose size is larger than 20000 bytes)
sizeKB	File size in Kilobytes (KiB)	put c:\folder\ mybucket -cond:"sizeKB > 200" (upload all files in c:\folder\ whose size is larger than 200 Kilobytes)
sizeMB	File size in Megabytes	put c:\folder\ mybucket -cond:"sizeMB < 100" (upload all files in c:\folder\ whose size is smaller than 100 Megabytes)
md5 (or etag)	MD5 of the file (also referred to as 'etag')	put c:\folder\ mybucket -cond:"md5 <> s3_md5" (upload all files in c:\folder\ whose MD5 value , i.e. etag, is different from the MD5 value, i.e. etag, of the corresponding file on S3)

S3 OBJECT VARIABLES (USABLE IN ALL FILTER CONDITIONS)

File Time / Date Variables	Explanation	Example
s3_day	The S3 object's day of the month, in the range 1 through 31	ls mybucket\folder\ -cond:"s3_day = 15" (list all files in c:\folder\ that were lastly modified on the 15th day of the month)
s3_month	The S3 object's month of the year, in the range 1 through 12 (1 = January, 12 = December)	del mybucket -s -cond:"s3_month = 1" (recursively delete all files in bucket 'mybucket' that were lastly modified in a January)
s3_year	The S3 object's year, 1970 to 2038	ls mybucket -s -cond:"s3_year <> 2012" (list all files in mybucket that were not lastly modified in 2012)
s3_date	The S3 object's date: 'YYYY-MM-DD'	s3_date = '2012-04-03' (true if S3 file's date is '2012-04-03') s3_date matches '2012-04-*' (true if S3 file's date is in April 2012) Note that <i>s3_date</i> is a text value and values must be surrounded by two '
s3_time	The S3 object's time: 'HH:MM:SS'	s3_time matches '08:12:10' (true if file's time matches '08:12:10')
s3_hour	The S3 object's hour, in the range 0 through 23	ls mybucket -s -cond:"s3_hour > 8" (list all files in mybucket that were lastly modified after 8)

s3_minute	The S3 object's minute, in the range 0 through 59	ls mybucket -s -cond:"s3_minute < 30 " (list all files in mybucket that were lastly modified in the first half of the hour)
s3_second	The S3 object's second, in the range 0 through 59	del mybucket -s -cond:"s3_second % 2 = 0" (recursively delete all files in mybucket that were lastly modified on an even second)
s3_dayofweek	The S3 object's day of the week (0 = Sunday, 1 = Monday, etc. to 6 = Saturday)	del mybucket -s -cond:"s3_dayofweek = 1" (recursively delete all files in bucket 'mybucket' that were lastly modified on a Monday)
s3_dayofyear	The S3 object's day of the year. Number from 1 to 366	ls mybucket -s -cond:"s3_dayofyear = 100" (recursively list all files in bucket 'mybucket' that were lastly modified on the 100th day of a year)
s3_weeknumber	The S3 object's week number (ISO 8601)	ls mybucket -s -cond:"s3_weeknumber = 22" (recursively list all files in bucket 'mybucket' that were lastly modified on the 22nd week of a year)
s3_endofmonth	The S3 object's end of month's date: 'YYYY-MM-DD'	s3_endofmonth = "2012-04-30" (true if S3 file's timestamp is in April 2012) s3_endofmonth matches "*-*-30" (true if S3 file's timestamp is in a month with 30 days)
s3_age_months	The S3 object's age in months.	ls mybucket\folder*.txt -cond:"s3_age_months > 6" (list all S3 files with extension txt in mybucket\folder\ that are older than 6 months)
s3_age_days	The S3 object's age in days.	ls mybucket\folder*.txt -cond:"s3_age_days < 31" (list all S3 files with extension txt in mybucket\folder\ that are newer than 31 days)
s3_age_hours	The S3 object's age in hours.	ls mybucket -cond:"s3_age_hours > 1" (list all S3 files in mybucket that are older than 1 hour)
s3_age_mins	The S3 object's age in minutes.	del mybucket\folder\ -cond:"s3_age_mins < 30" (delete all S3 files in mybucket\folder\ that are newer than 30 minutes)
s3_age_secs	The S3 object's age in seconds.	ls mybucket\folder*.txt -cond:"s3_age_secs > 90" (list all S3 files with extension txt in mybucket\folder\ that are older than 90 seconds)
s3_name	S3 object's name	ls mybucket\folder*.txt -cond:"s3_name regex_matches '^A(.*) ^B(.*)'" (list S3 txt files that have file name starting with 'A' or 'B', match is case sensitive, using regular expressions. Use (?i) marker, e.g. (?i)^A(.*) ^B(.*) to make the regular expression case-

		insensitive)
s3_path	S3 object's path (includes folder)	ls *.htm -s -cond:"s3_path contains '/' subfol/" (list all S3 htm files in all buckets that are in a subfolder called 'subfol')
s3_extension	S3 object's extension	ls -s -cond:"s3_extension = '.htm'" (list all S3 htm files in all buckets)
s3_stem	S3 object's stem (name without extension)	ls -s -cond:"s3_stem = 'check'" (list all S3 files with stem matching check. Stem = file name without extension. Operator = is case sensitive)
s3_size	S3 object's size in bytes	ls mybucket -s -cond:"s3_size>0" (list all S3 files in bucket <i>mybucket</i> that are larger than 0 bytes) ls mybucket -s -cond:"s3_size=0" (list all empty S3 files in bucket <i>mybucket</i>)
s3_sizeKB	S3 object's size in Kilobytes (KiB)	ls mybucket -s -cond:"s3_sizeKB>1000" (list all S3 files in bucket <i>mybucket</i> that are larger than 1000 Kilobytes)
s3_sizeMB	S3 object's size in Megabytes	ls mybucket -s -cond:"s3_sizeKB>100" (list all S3 files in bucket <i>mybucket</i> that are larger than 100 Megabytes)
s3_md5 (or s3_etag)	S3 object's MD5 (also referred to as 'etag')	put * mybucket -s -cond:"etag <> s3_etag" (upload all files in current folder and subfolders to bucket <i>mybucket</i> if their etag is different from the corresponding S3 etag)
s3_owner	S3 object's owner	del mybucket -s -cond:"s3_owner = 'Marc'" (delete all files in <i>mybucket</i> whose owner is 'Marc')
s3_owner_id	S3 object's owner ID	ls mybucket -s -cond:"s3_owner_id = '1234ABCD'" (list all files in <i>mybucket</i> whose owner ID is "1234ABCD")
s3_storage_class	S3 object's storage class	ls mybucket -s -cond:"s3_storage_class = 'STANDARD'" (list all files in <i>mybucket</i> with storage class 'STANDARD')
s3_is_directory	True if S3 object is a directory, false otherwise	ls mybucket -s -cond:"s3_is_directory = true" (list all directories in <i>mybucket</i>)
s3_exists	True if S3 object exists, false otherwise	put * mybucket -s -cond:"s3_exists = false" (upload all files in current folder and subfolders to S3 bucket <i>mybucket</i> only if they do not already exist on S3)

s3_version_id	S3 object's version ID	Is mybucket -s -cond:"s3_version_ID <> " " (list all objects in mybucket that have a version ID)
s3_is_latest_version	True if an object is the latest version of an object in a versioning enabled bucket	Is mybucket -s -cond:"s3_is_latest_version = true" (list all latest versions of objects in mybucket. This requires mybucket to be versioning enabled) Is mybucket -s -cond:"s3_is_latest_version = false" (list all previous versions of objects. This requires mybucket to be versioning enabled) delete mybucket/* -s -cond:"s3_is_latest_version = false" (delete all previous versions of all objects in mybucket. This requires mybucket to be versioning enabled)
s3_prev_version_number	Contains the previous version number, e.g. the last previous version of an object is the number 1, then 2, 3, etc.	del mybucket/* -s -cond:"s3_prev_version_number > 3" (delete previous versions of an object from the fourth version higher. If an object has only 1, 2 or 3 previous versions, then those previous versions are not deleted. This requires mybucket to be versioning enabled)
s3_is_delete_marker	True if object is a delete marker in a versioning enabled bucket	Is mybucket -s -cond:"s3_is_delete_marker = true" (list all delete markers in mybucket. This requires mybucket to be versioning enabled)
cache-control	S3 object's cache-control header	Is mybucket -s -cond:"cache_control = "" (list all files in mybucket that do not have a cache-control header)
s3_object_max_age	The S3 object's max-age value as specified in the cache-control header	Is mybucket -s -cond:"max-age > 0 " (list all files in mybucket that have max-age in the cache-control header greater than zero)
content-type	S3 object's content-type header	Is mybucket -s -cond:"content-type = 'text/html'" (list all files in <i>mybucket</i> that have the content-type header set to 'text/html')
x-amz-server-side-encryption	S3 object's encryption header	Is mybucket -s -cond:"x-amz-server-side-encryption = 'AES256'" (list all files in <i>mybucket</i> that have the x-amz-server-side-encryption header set to 'AES256')
x-amz-website-redirect-location	S3 object's redirect location header	Is mybucket -s -cond:"x-amz-website-redirect-location = '/page2.htm'" (list all files in <i>mybucket</i> that have the x-amz-website-redirect-location header set to '/page2.htm')
x-amz-version-id	S3 object's version ID header	Is mybucket -s -cond:"x-amz-version-id = 'value'" (list all files in <i>mybucket</i> that have the x-amz-version-id header set to 'value')

x-amz-expiration	S3 object's expiration header	Is mybucket -s -cond:"x-amz-expiration = 'value'" (list all files in <i>mybucket</i> that have the x-amz-expiration header set to 'value')
s3_object_expires	True if S3 objects has an expiration x-amz-expiration header, e.g. it expires	Is mybucket -s -cond:"s3_object_expires = true" (list all files in mybucket that are set to expire at some date in the future)
expiry_date, expiry_year, expiry_month, expiry_day, expiry_dayofweek, expiry_dayofyear, expiry_endofmonth, expiry_weeknumber, expiry_hour, expiry_minute, expiry_second, expiry_time, expiry_timestamp	If a S3 object has an expiry time and date set in the x-amz-expiration header, then these values contain the expiry date (format: YYYY-MM-DD), year (YYYY), month (1 to 12), day (1 to 31), day of week (0 = Sunday, 1 = Monday, etc. to 6 = Saturday), day of year (1 to 366), end of month (YYYY-MM-DD), week number (ISO 8601), hour (0 to 23), minute (0 to 59), second (0 to 59), time (HH:MM:SS) and timestamp (total seconds since epoch)	Is mybucket -s -cond:"expiry_weeknumber = 10" (list all objects in mybucket that will expire in week 10) Is mybucket -s -cond:"expiry_month = 5 and expiry_year = 2015" (list all objects in mybucket that will expire in May 2015) Is mybucket -s -cond:"(expiry_month > 5 and expiry_year = 2015) or expiry_year > 2015" (list all objects in mybucket that will expire after May 2015)
expiry_months, expiry_days, expiry_hours, expiry_mins, expiry_secs	If a S3 object has an expiry time and date set in the x-amz-expiration header, then these values contain the number of months, days, hours, minutes and seconds until the object expires	Is mybucket -s -cond:"expiry_months < 10" (list all objects in mybucket that will expire in less than 10 months) Is mybucket -s -cond:"expiry_days > 100" (list all objects in mybucket that will expire in more than 100 days)
x-amz-restore	S3 object's restore header	Is mybucket -s -cond:"x-amz-restore = 'value'" (list all objects in mybucket that have the x-amz-restore header set to 'value')
glacier_restored	True if a Glacier object is restored or being restored, e.g. the x-amz-restore header is not empty	Is mybucket -s -cond:"glacier_restored = true" (list all Glacier objects in mybucket that are restored)
restore_ongoing_request	True if a Glacier object is currently being restored to S3 after a restore request	Is mybucket -s -cond:"restore_ongoing_request = true" (list all Glacier objects in mybucket that are currently being restored to S3)
restore_expiry_date, restore_expiry_year, restore_expiry_month,	If a restored S3 Glacier object has a restore expiry time and date set in the x-amz-restore header, then these values contain the restore expiry date (format: YYYY-MM-DD), year (YYYY), month (1 to 12), day (1 to 31), day of week (0 =	Is mybucket -s -cond:"restore_expiry_month = 12 and expiry_year = 2015" (list all restored Glacier objects in mybucket that will expire in December 2015) Is mybucket -s -cond:"(restore_expiry_month > 5 and restore_expiry_year = 2015) or

restore_expiry_day, restore_expiry_dayofweek, restore_expiry_dayofyear, restore_expiry_endofmonth, restore_expiry_weeknumber, restore_expiry_hour, restore_expiry_minute, restore_expiry_second, restore_expiry_time, restore_expiry_timestamp	Sunday, 1 = Monday, etc. to 6 = Saturday), day of year (1 to 366), end of month (YYYY-MM-DD), week number (ISO 8601), hour (0 to 23), minute (0 to 59), second (0 to 59), time (HH:MM:SS) and timestamp (total seconds since epoch)	restore_expiry_year > 2015" (list all Glacier restored objects in mybucket that will expire after December 2015)
restore_expiry_months, restore_expiry_days, restore_expiry_hours, restore_expiry_mins, restore_expiry_secs	If a restored S3 Glacier object has a restore expiry time and date set in the x-amz-expiration header, then these values contain the number of months, days, hours, minutes and seconds until the object restore expires	Is mybucket -s -cond:"restore_expiry_days < 1" (list all restored Glacier objects in mybucket that will expire in less than 1 day) Is mybucket -s -cond:"restore_expiry_days > 100" (list all restored Glacier objects in mybucket that will expire in more than 100 days)
x-amz-meta-*	S3 object's custom metadata header	Is mybucket -s -cond:"x-amz-meta-mycustomheader = 'myvalue'" (list all files in <i>mybucket</i> that have the x-amz-meta-mycustomheader header set to "myvalue")
s3_acl	S3 object's acl. Different users are separated by	Is mybucket -s -cond:"s3_acl contains 'marc'" (list all files in <i>mybucket</i> that have an ACL set for 'marc')
s3_acl_is_private	True if S3 object's acl is set to private, false otherwise	Is mybucket -s -cond:"s3_acl_is_private = true" (list all private files in <i>mybucket</i>)
s3_acl_is_public_read	True if S3 object's acl is set to public read, false otherwise	Is mybucket -s -cond:"s3_acl_is_public_read = true" (list all public-read files in <i>mybucket</i>)
s3_acl_is_public_read_write	True if S3 object's acl is set to public read and write, false otherwise	Is mybucket -s -cond:"s3_acl_is_public_read_write = true" (list all public-read-write files in <i>mybucket</i>)
s3_acl_*	S3 object's specific user acl	Is mybucket -s -cond:"s3_acl_marc contains =

		'write'" (list all files in <i>mybucket</i> which mark can write)
--	--	---

CURRENT TIME VARIABLES (USABLE IN ALL FILTER CONDITIONS)

Time Variables	Explanation	Example
curr_day	The current day of the month, in the range 1 through 31	curr_day = 15 OR curr_dayofweek = 1 (true on Mondays or on the 15th of every month).
curr_month	The current month of the year, in the range 1 through 12 (1 = January, 12 = December)	curr_month = 1 (true in January).
curr_year	The current year, 1970 to 2038	curr_year <> 2012 (true if the current year is not 2012).
curr_date	Today's date: "YYYY-MM-DD"	curr_date = '2013-04-03' (true if date is 2013-04-03) curr_date matches '2012-04-*' (true in April 2012)
curr_time	Today's time: "HH:MM:SS"	curr_time matches '08:1*:*' (true if the current time is in the first 10 minutes of the 8th hour)
curr_hour	The current hour, in the range 0 through 23	curr_hour >= 0 AND curr_hour <= 7 (true between midnight and 7:59 am).
curr_minute	The current minute, in the range 0 through 59	curr_minute >= 30 (true in the second half of each hour).
curr_second	The current second, in the range 0 through 59	curr_second >= 30 (true in the second half of each minute).
curr_dayofweek	Current day of the week (0 = Sunday, 1 = Monday, etc. to 6 = Saturday)	curr_dayofweek = 0 (true on Sundays). curr_dayofweek = 0 OR curr_dayofweek = 6 (true on Sundays and Saturdays). curr_dayofweek <> 1 (true if day of week is not Monday).
curr_dayofyear	Current day of the year. Number from 1 to 366	curr_dayofyear = 32 (true on the 1st of February) curr_dayofyear % 2 = 0 (true on even days)
curr_weeknumber	Current week number (ISO 8601)	curr_weeknumber = 32 (true on the 32nd week of the year)
curr_endofmonth	Current end of month's date: "YYYY-MM-DD"	curr_endofmonth = '2012-04-30' (true in April 2012) curr_endofmonth matches '*-*-30' (true if today's month has 30 days)

SPECIAL FUNCTIONS (USABLE IN ALL FILTER CONDITIONS)

Function	Explanation	Example
<code>get_env(VariableName)</code>	Returns the text value of the environment variable <i>VariableName</i>	<code>get_env('username') = 'administrator'</code>
<code>to_lower(Text)</code>	Converts <i>Text</i> to lowercase	<code>ls -s -cond:"to_lower(s3_extension) = '.htm'"</code> (list all S3 htm files in all buckets, also if they have extension in uppercase .HTM)
<code>is_lowercase(Text)</code>	Returns true if <i>Text</i> is lower case, false otherwise	<code>ls -s -cond:"is_lowercase(s3_path) = false"</code> (list all objects in all buckets that are not lower case)
<code>pad(Text, Length, Char)</code>	Left pads <i>Text</i> with <i>Char</i> up to <i>Length</i>	<code>pad(s3_month, 2, '0')</code> returns <i>s3_month</i> with 0 padding, i.e. 1 becomes 01, 2 becomes 02, etc, but 12 remains 12.
<code>concat(Text1, Text2)</code>	Concatenates two text strings into one	<code>concat(concat(s3_year, pad(s3_month, 2, '0')), pad(s3_day, 2, '0'))</code>
<code>mid(Text, Start, End)</code>	Returns part of <i>Text</i> between <i>Start</i> and <i>End</i>	<code>mid('Sydney', 3, 2)</code> returns 'dn' <code>mid('Sydney', 1, 5)</code> returns 'Sydne' <code>mid('Sydney', 4, 100)</code> returns 'ney'
<code>value(Text)</code>	Returns the numerical value of <i>Text</i>	<code>value(mid('max-age=2000',9,0)) > 1999</code> returns true <code>value('2000')</code> returns 2000
<code>extract(Text, Regex)</code>	Extracts part of text from <i>Text</i> using regular expression <i>Regex</i>	<code>extract('Max-age=2000', 'max-age *= *(.*)')</code> returns '2000'
<code>extract_value(Text1, Text2)</code>	Extracts <i>Text2</i> value from <i>Text1</i>	<code>extract_value('max-age=2000', 'max-age')</code> returns 2000 <code>extract_value('private, max-age=20, no-cache', 'max-age')</code> returns 20

25 Command Shortcuts

Command shortcuts are useful to map long commands to short tags and make it easy to re-issue the same command again and again.

For instance, if you use S3Express to backup your files to S3, instead of having to retype a long command each time, you can simply define a command shortcut once and use it as needed, like this:

```
c1 put c:\folder\ mybucket -onlydiff
```

This assigns the command 'put c:\folder\ mybucket -onlydiff' to shortcut c1.

Next time, instead of having to type *put c:\folder\ mybucket -onlydiff* just type *c1* at the prompt and S3Express will issue the command *put c:\folder\ mybucket -onlydiff* for you.

Command shortcuts are customizable and will be remembered each time S3Express starts. Once a shortcut is assigned you can re-use it also the next time S3Express starts.

You can re-assign shortcuts or reset them when they are no longer needed.

S3Express supports up to 9 command shortcuts, c1, c2, c3, ... to c9.

c1, c2, c3 ... c9 (execute memorized command)

c1 COMMAND, c2 COMMAND ... c9 (assign COMMAND to shortcut c1, c2, ...)

c1 <last>, c2 <last> ... c9 (assign the last executed command to shortcut c1, c2, ...)

c1 <reset>, c2 <reset> ... c9 (reset shortcut c1, c2, ...)

c1 +, c2 +, ... c9 + (recall memorized command)

cshow (show all assigned shortcuts)

Examples

c1 put "c:\folder A\" mybucket -onlydiff -keep (assign command 'put "c:\folder A\" mybucket -onlydiff -keep' to shortcut c1. Then just type c1 to issue command 'put "c:\folder A\" mybucket -onlydiff -keep')

c1 (execute shortcut command c1. This will execute 'put "c:\folder A\" mybucket -onlydiff -keep')

c1 <last> (assign the last executed command to shortcut c1)

c1 <reset> (reset the c1 shortcut, c1 will no longer map to any command)

c5 + (recall memorized command c5 to command line)

cshow (show all assigned shortcuts)

26 Command Variables

Command variables are variables that can be used with all S3Express commands. Variables are substituted with the real values just before executing a command. Variables must be surrounded by `<*>` and `*>`.

COMMAND VARIABLES (USABLE IN ALL COMMANDS)

Variables	Explanation
<code><*year*></code>	Current year
<code><*month*></code>	Current month
<code><*day*></code>	Current day
<code><*date*></code>	Current date, equivalent to <code><*year*><*month*><*day*></code>
<code><*hour*></code>	Current hour, range 0 through 23
<code><*minute*></code>	Current minute, range 0 through 59
<code><*second*></code>	Current second, range 0 through 59
<code><*dayofweek*></code>	Current day of the week (0 = Sunday, 1 = Monday, etc. to 6 = Saturday)
<code><*dayofyear*></code>	Current day of the year, number from 1 to 366
<code><*weeknumber*></code>	Current week number (ISO 8601)
<code><*time*></code>	Current time, equivalent to <code><*hour*><*min*><*sec*></code>
<code><*computer*></code>	Name of computer where S3Express is running
<code><*user*></code>	Name of user running S3Express
Windows Environment Variables: <code><*environment_variable*></code>	<p><u>Any set Windows Environment Variable can be used as a command variable, but '<*>' and '*>' must be used instead of '%'</u>.</p> <p>For example:</p> <p><code>%ALLUSERSPROFILE%</code> becomes <code><*ALLUSERSPROFILE*></code></p> <p><code>%COMPUTERNAME%</code> becomes <code><*COMPUTERNAME*></code></p> <p><code>%HOMEPATH%</code> becomes <code><*HOMEPATH*></code></p> <p><code>%WINDIR%</code> becomes <code><*WINDIR*></code></p> <p>etc.</p>

Command Variables Examples:

Upload files to a different subfolder in bucket 'mybucket' every day of year (rotating):

```
put c:\folder\ mybucket/<*dayofyear*> -s
```

<*dayofyear*> is replaced with the current day of the year, number from 1 to 366, depending on when S3Express is running

Upload files to a different subfolder in bucket 'mybucket' based on computer name running S3Express:

```
put c:\folder\ mybucket/<*computer*> -s
```

<*computer*> is replaced with the name of the computer running S3Express

List all files in a S3 bucket lastly modified today:

```
ls mybucket -s -cond="s3_dayofyear = <*dayofyear*> and s3_year = <*year*>"
```

See [Filter Conditions](#) for detailed explanation of the -cond parameter.

Delete all files in a S3 bucket that were lastly modified yesterday:

```
del mybucket -s -cond="(s3_dayofyear = <*dayofyear*> - 1) and s3_year = <*year*>"
```

See [Filter Conditions](#) for detailed explanation of the -cond parameter.

27 Multipart Uploads

In order to make it faster and easier to upload larger objects, Amazon S3 has introduced the new **multipart upload** feature and S3Express fully supports this feature.

By specifying the flag **-mul** when uploading files with the command [put](#), S3Express will break your files into chunks and upload them separately. You can instruct S3Express to upload a number of chunks in parallel using the flag **-t**. If the upload of one single chunk fails, you can simply restart the upload and S3Express will restart from the last successful chunk instead of having to re-upload the entire file. If you do not want to restart an unfinished multipart upload, use the command [rmup!](#) to remove the upload.

Note: S3Express will also automatically apply the **correct MD5** value when uploading files using multipart uploads: many S3 tools are unable to do that.

28 Scripting via Command Line

When you start S3Express, the S3Express shell opens with a command prompt where you can type and execute all commands or copy and paste them, as shown in this picture:

```

S3Express 1.0 (Build 1000) 64-bit (SN5) windows Server :: TGRMN Software
Home Page: www.s3express.com

:: Type 'h', 'help', 'htmlhelp' or 'pdfhelp' for help.
:: Type 'h <command> for help on a specific command, e.g. h put
:: Type 'q', 'quit' or 'exit' to exit.

S3 Authorization Loaded.
Name           : <noname>
Access Key ID  : AKIAJVP4DI6I
Secret Access Key : AQhjCmNPqr...

/>_

```

But alternatively you can also run S3Express in automated mode, by passing commands to execute on the S3Express command line or via a text file.

The **S3Express command line** supports the following flags:

S3Express [commands, ...] [-exit] [-ini:commandfile] [-od] [-nm] [-h]

Command Line Flag	Description	Example
[commands, ...]	A list of commands. Separate commands with spaces. If needed, surround each command with double quotation marks (").	S3Express ls S3Express "put c:\folder* bucket -onlydiff" "ls bucket" q S3Express "put c:\folder* bucket" "put c:\folderA*.jpg bucket/jpgfolder/" q
[-exit]	Automatically exit S3Express once all the commands are executed. This is equivalent to having the command q as the last command.	S3Express "put c:\folder* bucket -onlydiff"
[-ini:commandfile]	Load commands from file <i>commandfile</i> . Each command should be on a separate line. The file should be a text file saved as UTF-8 or ANSI , see also the command exec .	S3Express -ini:"c:\folder\mycommandfile.txt" The file c:\folder\mycommandfile.txt must be a UTF-8 or ANSI text file. Each command must be on a separate line, e. g.:

		<pre>put c:\folder* bucket -onlydiff ls bucket q</pre> <p>More examples here: exec</p>
[-od]	Load commands from file S3Express.ini that is saved in the same folder as S3Express.exe.	<pre>S3Express -od</pre> <p>The file S3Express.ini must be a UTF-8 or ANSI text file. Each command must be on a separate line, see example above.</p>
[-nm]	Do not maximize the S3Express window at startup.	<pre>S3Express -nm</pre>
[-h]	Run S3Express completely <i>hidden</i> . No console is shown and no user interaction is possible. Note that the -h implies the -exit flag, that is, if the -h is used S3Express will automatically close once all the commands are executed.	<pre>S3Express "put c:\folder* bucket -onlydiff"</pre>

Note:

When using double quotation marks inside double quotation marks on the command line, the internal double quotation marks must be escaped by a \.

For example:

```
s3express "put \"g:\folder\folder with space\" \"bucket\folder with space\" -s -onlydiff -t:16 -minoutput -mul:50" -nm -exit
```

Alternatively the commands can be put in a text file (e.g. commandfile.txt):

```
put "G:\folder\folder with space" "bucket\folder with space" -s -onlydiff -t:16 -minoutput -mul:50
quit
```

and then run with : `s3express -ini:commandfile.txt -nm`

Exit Codes

S3Express.exe returns 0 if all executed commands were successful. It will return 1 otherwise.

29 Exit Codes

S3Express.exe returns 0 if all executed commands were successful. It will return 1 otherwise.

30 Redirect Command Output to a File

All S3Express commands support the parameter **-tofile:filename** to redirect the output from the screen to a file.

When the -tofile parameter is used, the output of a command is redirected from the screen to a file. If the file already exists, it is overwritten. For example:

```
ls mybucket -s -tofile:c:\output.txt
```

```
put c:\folder\ mybucket -tofile:x:\folder\output.txt
```

If no file path is specified, but only the file name, then the file is created in the current directory:

```
ls mybucket -s -tofile:output.txt
```

To append the output to an existing file, add a '+' at the end of the file name, e.g.:

```
ls mybucket -s -tofile:output.txt+
```

Alternatively, you can also redirect the whole program output to a file using the command line, e.g.

```
S3Express "put c:\folder\* bucket -onlydiff" "ls bucket" q > output.txt
```

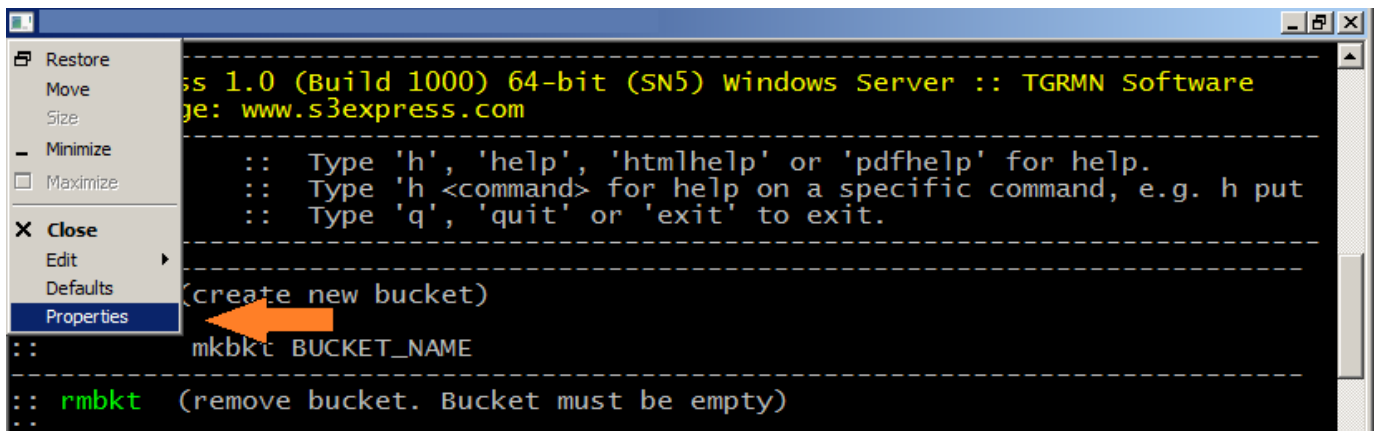
or to append to an existing file:

```
S3Express "put c:\folder\* bucket -onlydiff" "ls bucket" q >> output.txt
```

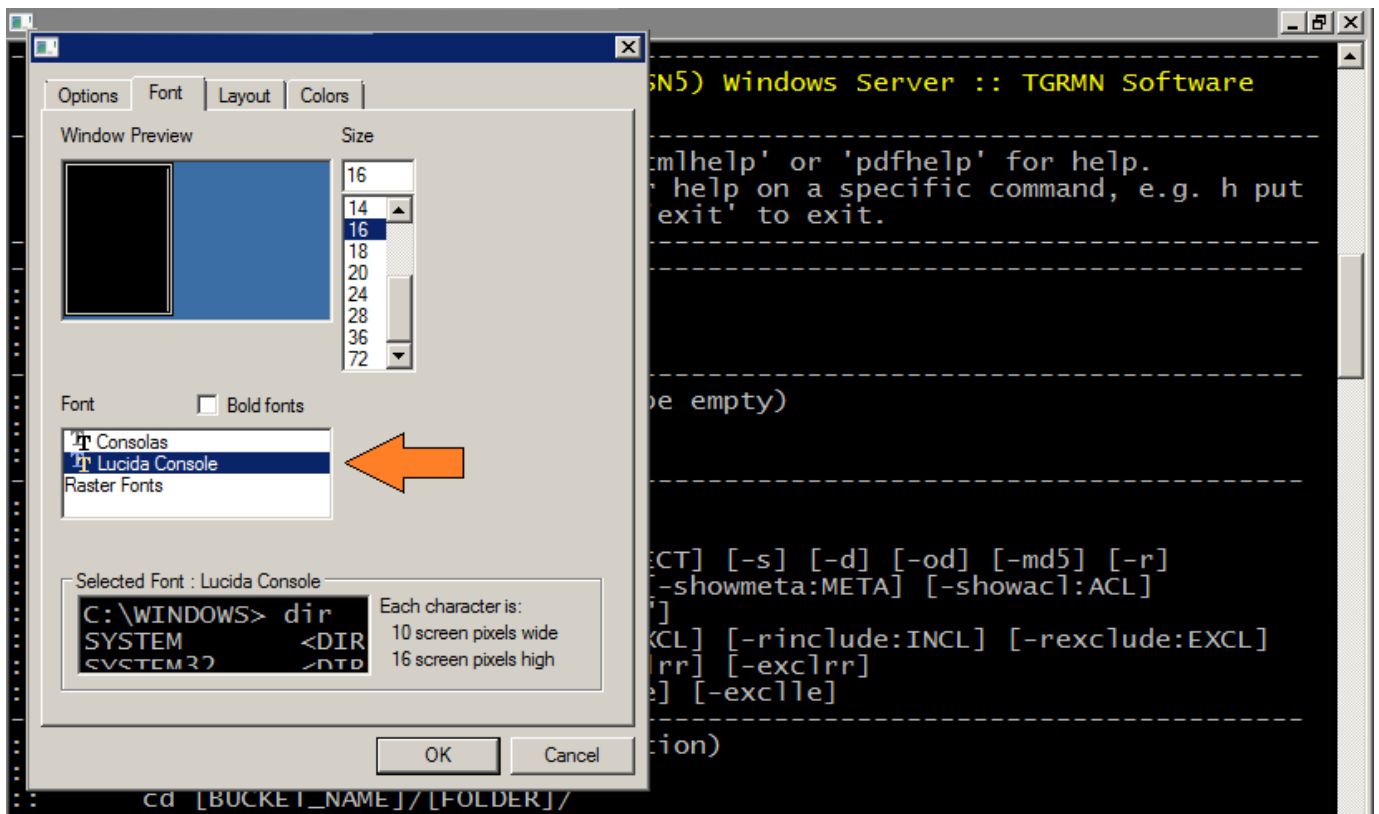
31 Unicode Support (åäö etc.)

Amazon S3™ and S3Express support Unicode and all alphabets in the world. In order to properly show all Unicode characters in Windows, you'll need to set the Windows console font to '**Lucida Console**', as shown below. This only has to be done once.

Step 1: Select 'Properties' from the console menu.



Step 2: Select the 'Lucida Console' font from the 'Font' tab.



32 Examples

Following are some examples showing S3Express functionality.

Buckets

Create a bucket.

```
mkbkt mybucket
```

Remove a bucket (bucket must be empty).

```
rmbkt mybucket
```

Empty a bucket (remove all objects within bucket).

```
del mybucket/* (or cd mybucket followed by del *)
```

List Objects

List all buckets.

```
ls
```

List all objects in 'mybucket'.

```
ls mybucket (or cd mybucket followed by just ls)
```

List all objects in 'mybucket' including in all subfolders.

```
ls mybucket -s (or cd mybucket followed by just ls -s)
```

List all objects in 'mybucket', including in all subfolders, that have .txt extension. Include MD5 values in the output.

```
ls mybucket/*.txt -s -md5
```

List all objects in 'mybucket', including in all subfolders, that have .txt extension. Include MD5 values, metadata and ACL in the output, in extended format (-ext). Note that showing metadata and/or ACL is slower as each object must be queried separately.

```
ls mybucket/*.txt -s -md5 -showmeta -showacl -ext
```

List all objects in 'mybucket', including in all subfolders, that are server-side encrypted.

```
ls mybucket -s -inclenc
```

List all objects in 'mybucket', including in all subfolders, that are server-side encrypted.

```
ls mybucket -s -inclenc
```

List all objects in 'mybucket', but not in subfolders, that have the header cache-control:max-age set to 60. Show metadata in the output (-showmeta).

```
ls mybucket -cond:"extract_value(cache-control,'max-age')=60" -showmeta
```

List all objects in 'mybucket', subfolders 'mysubfolder', that do not have the cache-control header. Show metadata in the output (-showmeta).

```
ls mybucket/mysubfolder/ -cond:"cache-control="" -showmeta
```

List all objects in 'mybucket', subfolders 'mysubfolder', that have the cache-control header. Show metadata in the output (-showmeta).

```
ls mybucket/mysubfolder/ -cond:"cache-control !=" " -showmeta
```

List all objects in 'mybucket', subfolders 'mysubfolder', that are larger than 5 Megabytes.

```
ls mybucket/mysubfolder/ -cond:"s3_sizeMB>5"
```

List all objects in 'mybucket', subfolders 'mysubfolder', that are larger than 5 Megabytes and have extension .txt or .gif or .jpg

```
ls mybucket/mysubfolder/ -cond:"s3_sizeMB>5" -include:"*.txt|*.jpg|*.gif"
```

List all objects in 'mybucket', including in all subfolders, that do not start with a, b or c (using regular expressions).

```
ls mybucket -s -rexclude:"^(a.|b.|c.)"
```

Note that in the example above -rexclude uses the object name to match. To match against the entire object path, use the s3_path variable, e.g.

```
ls mybucket -s -cond:"s3_path regex_matches '^(a.|b.|c.)' = false"
```

List all objects in 'mybucket', including in all subfolders, that are not 'private' ('private' means that owner gets FULL_CONTROL and no one else has access rights).

```
ls mybucket -s -cond:"s3_acl_is_private = false"
```

List all objects in 'mybucket', including in all subfolders, that are not 'public-read' ('public-read' means that owner gets FULL_CONTROL and the AllUsers group gets READ access).

```
ls mybucket -s -cond:"s3_acl_is_public_read = false"
```

List all objects in 'mybucket', including in all subfolders, and group output by object's extension.

```
ls mybucket -s -grp:ext
```

Show a summary of all objects in 'mybucket', including in all subfolders, which have the cache-control:max-age value greater than 0, and group the output by cache-control header value. Do not show each object, just a summary (-sum parameter).

```
ls mybucket -s -sum -grp:cache-control -cond:"extract_value(cache-control,'max-age')>0"
```

Put Objects (Uploads)

Upload all files that are in c:\folder\ to mybucket.

```
put c:\folder\ mybucket
```

Upload file c:\folder\file.txt to mybucket/subfolder

```
put c:\folder\file.txt mybucket/subfolder/
```

Upload files all *.txt that are in c:\folder\ to mybucket/subfolder/

```
put c:\folder\*.txt mybucket/subfolder/
```

Upload all files in c:\folder\ and its subfolders to mybucket using 3 parallel threads and multipart uploads. Throttle bandwidth to 50Kb/s. Make all uploaded files 'public-read' and set cache-control header to max-age=60.

```
put c:\folder\ mybucket -s -t:3 -mul -maxb:50 -cacl:public-read -meta:"cache-control:max-age=60"
```


Upload all files from `c:\folder\` to `mybucket` and apply S3 server-side encryption for all uploaded files.

```
put c:\folder\ mybucket -e
```

Upload all files from `c:\folder\` to `mybucket`. Before uploading, apply client-side local encryption.

```
put c:\folder\ mybucket -le
```

Upload non-empty files from `c:\folder\` to `mybucket` and keep metadata and ACL of files that are overwritten.

```
put c:\folder\ mybucket -cond:"size <> 0" -keep
```

Upload only changed or new files from `c:\folder\` to `mybucket` and keep metadata and ACL of files that are overwritten.

Changed files are files that have changed, that is they have different MD5 hash. New files are files that are not yet present on the S3 bucket.

Options `-onlynew` (upload only new files), `-onlynewer` (upload only files that have a newer timestamp) and `-onlyexisting` (re-upload only files that are already present on S3) are also available.

```
put c:\folder\ mybucket -onlydiff -keep
```

Upload only changed or new files from `c:\folder\` to `mybucket`. Purge (=delete) S3 files in `mybucket` that are no longer present in `c:\folder\`. Keep output to console to minimum (`-minoutput`).

```
put c:\folder\ mybucket -s -onlydiff -purge -minoutput
```

Upload all `*.jpg` and `*.gif` files from `c:\folder\` to `mybucket`, only if they are already existing on S3.

```
put c:\folder\ mybucket -include:*.jpg|*.gif -onlyexisting
```

Upload all `*.jpg` and `*.gif` files from `c:\folder\` to `mybucket`, only if files are already existing on S3. Simulation (`=preview`) only, shows list of files that would be uploaded.

```
put c:\folder\ mybucket -include:*.jpg|*.gif -onlyexisting -sim
```

Move all `*.jpg` and `*.gif` files from `c:\folder\` to `mybucket`.

```
put c:\folder\ mybucket -include:*.jpg|*.gif -move
```

Delete Objects / Copy Objects

Delete files in `mybucket`, including subfolders, that have `cache-control:max-age > 0` in the metadata.

```
del mybucket/* -s -cond:"extract_value(cache-control,'max-age') > 0"
```

Delete files in `mybucket`, including subfolders, that are empty.

```
del mybucket/* -s -cond:"size = 0"
```

Delete files in `mybucket`, including subfolders, with name starting with 'a'. Stop deleting files as soon as an error occurs.

```
del mybucket/* -s -cond:"name starts_with 'a'" -stoponerror
```

Delete previous versions of files in `mybucket`, which are older than 6 months, including subfolders.

```
del mybucket/* -s -onlyprev -cond:"s3_age_months>6"
```

Copy file `mybucket/myfile.txt` to `mybucket/subfolder/myfile.txt` and copy also metadata and ACL from source `mybucket/myfile.txt` to target `mybucket/subfolder/myfile.txt`.

```
copy mybucket/myfile.txt mybucket/subfolder/myfile.txt -keep
```

Metadata and Permissions (ACLs)

Set header cache-control:max-age=60 and x-amz-meta-test=yes to all files in mybucket/subfolder/.

```
setmeta mybucket/subfolder/* -meta:"cache-control:max-age=60|x-amz-meta-test=yes"
```

*Set header x-amz-meta-test=yes to all files in mybucket/subfolder/ that have extension *.exe or *.rpt.*

```
setmeta mybucket/subfolder/* -meta:"x-amz-meta-test=yes" -include:"*.exe|*.rpt"
```

*Set server-side encryption header (= encrypt files) to all files in mybucket/subfolder/ that are larger than 5MB and do not have extension *.exe or *.rpt.*

```
setmeta mybucket/* -e:+ -cond:"size_mb > 5" -exclude:"*.exe|*.rpt"
```

Get the metadata of the file.txt and show ALL metadata headers.

```
getmeta file.txt
```

Get metadata of the file.txt, but show only the cache-control header in the output.

```
getmeta file.txt -showmeta:"cache-control"
```

Get metadata of the file.txt, but show only the cache-control header and the x-amz-server-side-encryption header in the output.

```
getmeta file.txt -showmeta:"cache-control|x-amz-server-side-encryption"
```

Set canned ACL 'private' to all jpg files in mybucket, including in subfolders of mybucket.

```
setacl mybucket/*.jpg -s -acl:private
```

Set canned ACL 'public-read-write' to all txt files in mybucket, including in subfolders of mybucket

```
setacl mybucket/*.txt -s -acl:public-read-write
```

Grant read access to emailAddress=xyz@amazon.com and emailAddress=abc@amazon.com to all files in mybucket.

```
setacl mybucket/* -grant-read:"emailAddress=xyz@amazon.com, emailAddress=abc@amazon.com"
```

Get ACL of object.txt and show AllUsers permissions in the output.

```
getacl object.txt -showacl:allusers
```

List all objects in 'mybucket', including in all subfolders, that are not 'public-read' ('public-read' means that owner gets FULL_CONTROL and the AllUsers group gets READ access). The ls command is used.

```
ls mybucket -s -cond:"s3_acl_is_public_read = false"
```

Other

Save S3 authorization in S3Express:

```
saveauth FASWQSDSSSZXAS1SA AsFZEDy2BQfFSFzFfgKyyOF/xCaRcK4RMc
```

33 FAQ and Knowledge Base

Visit the S3Express FAQ and Knowledge Base online for more command line examples and tips.

FAQ and Knowledge Base:

 www.s3express.com/kb/

Most Viewed:

[How Do I Backup to Amazon S3 with S3Express?](#)

[PDF Tutorial 'Backup Files to Amazon S3 with S3Express'](#)

[Release History](#)

[Security Considerations](#)

[S3Express How-To](#)

34 How to Buy S3Express


You can buy S3Express licenses online on our website, using all major credit cards, paypal, check, or a bank transfer.

Buy Page:

 www.s3express.com/buy.htm

S3Express Homepage:

 www.s3express.com

 **Note:** The License Key and Text will be delivered via e-mail and you will then [enter them in S3Express](#) to unlock the trial.

35 How to Enter the License

Once you have received your **License Key** and **Text** via e-mail, you will need to enter it in S3Express.

1) Start S3Express

Start S3Express.exe

2) Use Command 'license' to enter the license

At the S3Express prompt use command [license](#) to enter the license key and text.

Type:

```
license "LICENSE_TEXT" "LICENSE_KEY"
```

where LICENSE_TEXT is your license text and LICENSE_KEY is your license key.

For example type:

```
license "Leoclara-Inc-USA" "213-12-111A8"
```

The license code will be validated and after a few seconds S3Express will be licensed to you. Please contact support@s3express.com should you have any difficulties licensing S3Express.

36 License Agreement

This S3Express Software License Agreement ("S3EXPSLA") is a legal agreement between you (either an individual or a single entity) and TGRMN Software for the S3Express software product, which includes computer software and associated media and printed materials, and may include "online" or electronic documentation ("SOFTWARE PRODUCT" or "SOFTWARE").

By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this S3EXPSLA. If you do not agree to the terms of this S3EXPSLA, uninstall the SOFTWARE PRODUCT and promptly return the unused SOFTWARE PRODUCT to the place from which you obtained it for a full refund.

This SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

Grant of License

This license agreement grants you the following rights:

- You may install and use one copy of the SOFTWARE PRODUCT on a single computer at a time and only by one user at a time.
- You may install and use copies of the SOFTWARE PRODUCT, up to, but not to exceed, the number of licenses shown on your purchase record.
- Each primary user of the SOFTWARE PRODUCT specified above, may also install and use an additional copy of the SOFTWARE PRODUCT on a portable device or home computer (not both), providing this copy is not used concurrently with the primary copy.
- Network Storage/Use - You may also store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT on your other computers over an internal network; however, you must acquire and dedicate a license for each separate computer on which the SOFTWARE PRODUCT is installed or run from the storage device.
- Concurrent Use - A license for the SOFTWARE PRODUCT may not be used concurrently on different computers.

Evaluation and Registration

This is not free software. You are hereby licensed to use this SOFTWARE PRODUCT for evaluation purposes without charge for a period of 21 days. If you use this SOFTWARE after the evaluation period, a registration fee is required. When payment is received you will be sent a license key. For information on ordering a license key visit www.s3express.com

Distribution

Provided that you verify that you are distributing the evaluation version, you are hereby licensed to make as many copies of the evaluation version of this SOFTWARE PRODUCT and documentation as you wish, give exact copies of the original evaluation version to anyone, and distribute the evaluation version of the SOFTWARE and documentation in its unmodified form via electronic means. There is no charge for any of the above.

You are specifically prohibited from charging or requesting donations for any such copies however made, and from distributing the SOFTWARE PRODUCT and/or documentation with other products (commercial or otherwise) without prior written permission.

Copyright and Trademark Rights

The SOFTWARE PRODUCT is owned by TGRMN Software. This Agreement does not grant you any intellectual property rights in the SOFTWARE PRODUCT.

Disclaimer of Warranty

THIS SOFTWARE PRODUCT AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. IN NO EVENT WILL TGRMN SOFTWARE BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS. Because of the various hardware and software environments into which S3Express may be put, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the SOFTWARE. ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

Restrictions

You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the SOFTWARE. You may not use, copy, modify or transfer copies of the SOFTWARE except as provided in this licence. You may not decompile, disassemble, or create derivative works based upon the SOFTWARE. You may not modify, adapt, translate, or create derivative works based upon the written documentation. You may not sub-license, rent, lease, sell or assign the SOFTWARE to others.

Governing Law

This Agreement shall be governed by, construed and enforced in accordance with the internal substantive laws (and not the laws of choice of laws) of South Australia, Australia, without giving effect to the conflict of laws provisions. Sole venue shall be in the applicable state and federal courts of South Australia.