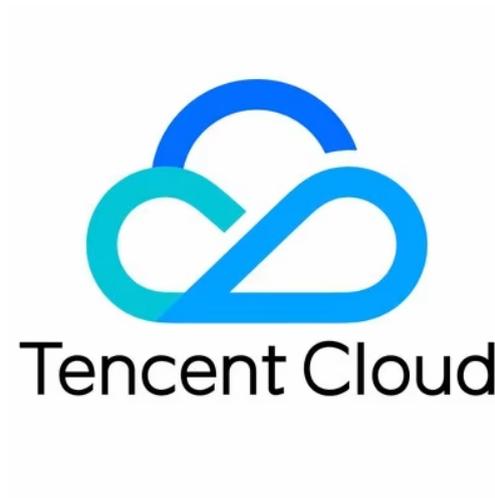


# Cloud Object Storage Tool Guide



## Copyright Notice

©2013–2026 Tencent Cloud. All rights reserved.

The complete copyright of this document, including all text, data, images, and other content, is solely and exclusively owned by Tencent Cloud Computing (Beijing) Co., Ltd. ("Tencent Cloud"); Without prior explicit written permission from Tencent Cloud, no entity shall reproduce, modify, use, plagiarize, or disseminate the entire or partial content of this document in any form. Such actions constitute an infringement of Tencent Cloud's copyright, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

## Trademark Notice



This trademark and its related service trademarks are owned by Tencent Cloud Computing (Beijing) Co., Ltd. and its affiliated companies ("Tencent Cloud"). The trademarks of third parties mentioned in this document are the property of their respective owners under the applicable laws. Without the written permission of Tencent Cloud and the relevant trademark rights owners, no entity shall use, reproduce, modify, disseminate, or copy the trademarks as mentioned above in any way. Any such actions will constitute an infringement of Tencent Cloud's and the relevant owners' trademark rights, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

## Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

## Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

# Contents

## Tool Guide

Tool Overview

Installation and Configuration of Environment

Hadoop

COSBrowser

COSBrowser Overview

User Guide for Desktop Version

User Guide for Mobile Version

Installation and Login

Mobile Version Features

Bucket Management and Operations

File Management and Operations

Resource Packs

COSCMD

COSCLI (Beta)

COSCLI Overview

Download and Installation Configuration

Common Options

Common Commands

Generating and Modifying Configuration Files – config

Creating Buckets – mb

Deleting Buckets – rb

Tagging Bucket – bucket-tagging

Querying Bucket/Object List – ls

Obtaining Statistics on Different Types of Objects – du

Uploading/Downloading/Copying Objects – cp

Syncing Upload/Download/Copy – sync

Deleting Objects – rm

Getting File Hash Value – hash

Listing Incomplete Multipart Uploads – lsparts

Clearing Incomplete Multipart Uploads – abort

Retrieving Archived Files – restore

Getting Pre-signed URL – signurl

Creating/Obtaining a Symbolic Link – symlink

Viewing Contents of an Object – cat

**Listing Contents and Statistics Under a Directory – lsdu**

**Bucket Versioning – bucket-versioning**

**Object Tag – object-tagging**

**Management Object ACL – object-acl**

**Managing Bucket ACL – bucket-acl**

**Bucket Policy – bucket-policy**

**Bucket Encryption Policy –bucket-encryption**

**Inventory – inventory**

**FAQs**

**COS Migration**

**FTP Server**

**Hadoop**

**COSDistCp**

**HDFS TO COS**

**GooseFS-Lite**

**Online Tools**

**COS Request Tool**

**Policy Generator**

**Self-Diagnosis Tool**

# Tool Guide

## Tool Overview

Last updated: 2023-09-20 10:37:38

Tool	Note
<a href="#">COS Browser</a>	This tool makes it easy for users to perform data upload/download, access link generation, and other operations in a visualized manner.
<a href="#">COSCLI</a>	COS provides the command-line client COSCLI to allow you to upload, download, delete, and perform other operations on COS objects by using simple commands.
<a href="#">COSCMD Tool</a>	This tool allows users to perform operations such as batch upload/download/deletion of objects by using simple command line instructions.
<a href="#">COS Migration Tool</a>	This tool is used to migrate data from multiple source data addresses (such as local server, and other cloud storage services) to COS.
<a href="#">FTP Server Tool</a>	This tool is used to upload/download files to/from COS via FTP server.
<a href="#">COSFS Tool</a>	In Linux, this tool is used to mount buckets to a local file system and operate objects in COS via the local file system.
<a href="#">Hadoop Tool</a>	Helps integrate COS with big data computing frameworks such as Hadoop, Spark, and Tez, so that they can read and write COS data.
<a href="#">COSDistCp</a>	COSDistCp is a MapReduce-based distributed file copy tool mainly used for data copy between HDFS and COS.
<a href="#">Hadoop-cos-DistChecker</a>	Verifies the directory integrity after you use the <code>hadoop distcp</code> command to migrate data from HDFS to COS.
<a href="#">HDFS TO COS Tool</a>	This tool is used to copy the data on HDFS to COS.
<a href="#">Online Auxiliary Tool</a>	Web-based COS tools, including <a href="#">COS signing tool</a> and <a href="#">COS request tool</a> .
<a href="#">Diagnostic Tool</a>	A web tool for troubleshooting error requests
<a href="#">Document Preview Integration Experience</a>	The Document Preview Access Experience Tool is a web-based tool provided by COS for users to facilitate the integration and debugging of document-to-HTML preview services.

## Tool

Of course, COS not only offers the above applications and services but also provides a variety of popular open-source applications integrated with Tencent Cloud COS plugins. Click [here](#) to launch and start using them immediately!

If you have any other tool requirements, please feel free to submit them [here](#). We will promptly evaluate your needs. Thank you!

## Upload Capabilities

The upload capabilities of different tools are detailed as below:

Tool	Simple Upload	Multipart upload	Checkpoint Restart	Advanced upload	Consistency Check	Pre-Signed URL Generation
COSBrowser	This feature is supported.	This feature is supported.	This feature is supported.	Supported. By default, multipart upload will be triggered for files of 8 MB or above.	Supported for MD5 check	Supported for file download.
COSCLI	This feature is supported.	This feature is supported.	This feature is supported.	Supported. The trigger threshold for multipart upload can be customized.	Supported for CRC64 check	This feature is supported.
COSCMD	This feature is supported.	This feature is supported.	This feature is supported.	By default, multipart upload is triggered when the file size exceeds 10 MB.	Supported for MD5 check	This feature is supported.
COS Migration	This feature is supported.	This feature is supported.	This feature is supported.	Supported. The trigger threshold for multipart upload can be	Supported for MD5 check	N/A

				customized.		
FTP Server	This feature is supported.	This feature is supported.	This feature is supported.	Supported. The trigger threshold for multipart upload can be customized.	Unavailable	N/A
COSFS	This feature is supported.	This feature is supported.	This feature is supported.	Supported. The trigger threshold for multipart upload can be customized.	Supported for MD5 check	N/A
Hadoop-COS	This feature is supported.	This feature is supported.	Not supported. There is an HDFS protocol conflict.	Supported. The trigger threshold for multipart upload can be customized.	Supported for CRC64 or CRC32 check	N/A
COSDist Cp Tool	This feature is supported.	This feature is supported.	Not supported. There is an HDFS protocol conflict.	Supported. The trigger threshold for multipart upload can be customized.	Supported for file size, CRC64, or CRC32 check	N/A

HDFS TO COS	This feature is supported.	This feature is supported.	Not supported. There is an HDFS protocol conflict.	Supported. The trigger threshold for multipart upload can be customized.	Supported for file name or size check	N/A
-------------	----------------------------	----------------------------	--	--	---------------------------------------	-----

**Note**

Advanced upload encapsulates simple and multipart uploads, allowing users to choose the upload method based on file size.

# Installation and Configuration of Environment Hadoop

Last updated: 2025-01-24 14:27:49

Hadoop tools rely on Hadoop 2.7.2 or later versions, enabling Tencent Cloud Object Storage (COS) to serve as the underlying storage filesystem for running upper-layer computational tasks. Hadoop clusters can be launched in three primary modes: standalone, pseudo-distributed, and fully distributed. This document focuses on setting up a fully distributed Hadoop environment using Hadoop 2.7.4 and provides a simple word count test demonstration.

## Preparation

- Prepare several servers.
- Install and configure the system by downloading from the [CentOS Official Website](#). This document uses CentOS 7.3.1611.
- Install the Java environment. For more information, see [Installing and Configuring Java](#).
- Install the Hadoop package: [Apache Hadoop Releases Download](#).

## Network Configuration

Use `ifconfig -a` to check the IP of each server, then use the ping command to check if they can ping each other, and record the IP of each server.

## Configuring CentOS

### Configure a Hostname

Set the corresponding hostname for each machine, such as "master", "slave\*", etc.

```
hostnamectl set-hostname master
```

### Configure hosts

```
vi /etc/hosts
```

Edit the content:

```
202.xxx.xxx.xxx master
202.xxx.xxx.xxx slave1
202.xxx.xxx.xxx slave2
202.xxx.xxx.xxx slave3
# Replace the IP address with the actual IP
```

## Turn off firewall

```
systemctl status firewalld.service    # Check the firewall status
systemctl stop firewalld.service      # Disable the firewall
systemctl disable firewalld.service    # Disable firewall startup at boot
```

## Time Synchronization

```
yum install -y ntp                    # Install the ntp service
ntpdate cn.pool.ntp.org               # Synchronize network time
```

## Install and configure JDK

Upload JDK installer package (such as `jdk-8u144-linux-x64.tar.gz`) to the `root` directory.

```
mkdir /usr/java
tar -zxvf jdk-8u144-linux-x64.tar.gz -C /usr/java/
rm -rf jdk-8u144-linux-x64.tar.gz
```

## Copy JDKs among hosts

```
scp -r /usr/java slave1:/usr
scp -r /usr/java slave2:/usr
scp -r /usr/java slave3:/usr
.....
```

## Configure environment variables for JDK of each host

```
vi /etc/profile
```

Edit the content:

```
export JAVA_HOME=/usr/java/jdk1.8.0_144
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

After saving the file, make `/etc/profile` take effect by executing the following command:

```
source /etc/profile # Make the configuration file effective
java -version # View java version
```

## Configure keyless access via SSH

Check the SSH service status on each host:

```
systemctl status sshd.service # Check SSH service
status
yum install openssh-server openssh-clients # Install SSH services;
skip this step if already installed.
systemctl start sshd.service # Start the SSH service.
If it's already installed, you don't need to perform this step.
```

Generate a key on each host:

```
ssh-keygen -t rsa # Generate key
```

On slave1:

```
cp ~/.ssh/id_rsa.pub ~/.ssh/slave1.id_rsa.pub
scp ~/.ssh/slave1.id_rsa.pub master:~/.ssh
```

On slave2:

```
cp ~/.ssh/id_rsa.pub ~/.ssh/slave2.id_rsa.pub
scp ~/.ssh/slave2.id_rsa.pub master:~/.ssh
```

And so on...

On the master:

```
cd ~/.ssh
cat id_rsa.pub >> authorized_keys
cat slave1.id_rsa.pub >>authorized_keys
cat slave2.id_rsa.pub >>authorized_keys
scp authorized_keys slave1:~/.ssh
scp authorized_keys slave2:~/.ssh
scp authorized_keys slave3:~/.ssh
```

## Installing and Configuring Hadoop

### Installing Hadoop

Upload the Hadoop installer package (such as `hadoop-2.7.4.tar.gz`) to the `root` directory.

```
tar -zxvf hadoop-2.7.4.tar.gz -C /usr
rm -rf hadoop-2.7.4.tar.gz
mkdir /usr/hadoop-2.7.4/tmp
mkdir /usr/hadoop-2.7.4/logs
mkdir /usr/hadoop-2.7.4/hdf
mkdir /usr/hadoop-2.7.4/hdf/data
mkdir /usr/hadoop-2.7.4/hdf/name
```

Go to the `hadoop-2.7.4/etc/hadoop` directory and proceed to the next step.

### Configure Hadoop

1. Modify the `hadoop-env.sh` file, adding the following content:

```
export JAVA_HOME=/usr/java/jdk1.8.0_144
```

If the SSH port is not the default 22, you can modify it in the `hadoop-env.sh` file:

```
export HADOOP_SSH_OPTS="-p 1234"
```

2. Modify `yarn-env.sh`

```
export JAVA_HOME=/usr/java/jdk1.8.0_144
```

3. Modify `slaves` configuration content:

```
Delete:
localhost
Add:
slave1
slave2
slave3
```

#### 4. Modify `core-site.xml`

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:/usr/hadoop-2.7.4/tmp</value>
  </property>
</configuration>
```

#### 5. Modify `hdfs-site.xml`

```
<configuration>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/usr/hadoop-2.7.4/hdf/data</value>
    <final>>true</final>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/usr/hadoop-2.7.4/hdf/name</value>
    <final>>true</final>
  </property>
</configuration>
```

#### 6. Copy `mapred-site.xml.template` and name it `mapred-site.xml`

```
cp mapred-site.xml.template mapred-site.xml
```

## 7. Modify `mapred-site.xml`

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>master:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>master:19888</value>
  </property>
</configuration>
```

## 8. Modify `yarn-site.xml`

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>master:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>master:8030</value>
  </property>
```

```
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:8031</value>
</property>
<property>
<name>yarn.resourcemanager.admin.address</name>
<value>master:8033</value>
</property>
<property>
<name>yarn.resourcemanager.webapp.address</name>
<value>master:8088</value>
</property>
</configuration>
```

## 9. Copy Hadoop among hosts

```
scp -r /usr/ hadoop-2.7.4 slave1:/usr
scp -r /usr/ hadoop-2.7.4 slave2:/usr
scp -r /usr/ hadoop-2.7.4 slave3:/usr
```

## 10. Configure Hadoop environment variables on each host

Open the configuration file:

```
vi /etc/profile
```

Edit the content:

```
export HADOOP_HOME=/usr/hadoop-2.7.4
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export HADOOP_LOG_DIR=/usr/hadoop-2.7.4/logs
export YARN_LOG_DIR=$HADOOP_LOG_DIR
```

Make the configuration file take effect:

```
source /etc/profile
```

## Start Hadoop

### 1. Format namenode

```
cd /usr/hadoop-2.7.4/sbin
hdfs namenode -format
```

## 2. Start

```
cd /usr/hadoop-2.7.4/sbin
start-all.sh
```

## 3. Check processes

If the master server includes ResourceManager, SecondaryNameNode, NameNode, etc., it indicates a successful startup, for example:

```
2212 ResourceManager
2484 Jps
1917 NameNode
2078 SecondaryNameNode
```

If processes on each slave contain DataNode and NodeManager, Hadoop is started successfully. For example:

```
17153 DataNode
17334 Jps
17241 NodeManager
```

## Running wordcount

The wordcount built in Hadoop can be called directly. After Hadoop starts, use the following command to work with files in HDFS:

```
hadoop fs -mkdir input
hadoop fs -put input.txt /input
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar
wordcount /input /output/
```

```
root@VM_96_24_centos /usr/hadoop-2.7.4# hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar wordcount /input /output/
17/07/18 23:04:51 INFO client.RMProxy: Connecting to ResourceManager at master/10.104.96.24:8032
17/07/18 23:04:53 INFO input.FileInputFormat: Total input paths to process : 1
17/07/18 23:04:53 INFO mapreduce.JobSubmitter: number of splits:1
17/07/18 23:04:54 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1500344813707_0002
17/07/18 23:04:54 INFO impl.YarnClientImpl: Submitted application application_1500344813707_0002
17/07/18 23:04:54 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy/application_1500344813707_0002/
17/07/18 23:04:54 INFO mapreduce.Job: Running job: job_1500344813707_0002
17/07/18 23:05:01 INFO mapreduce.Job: Job job_1500344813707_0002 running in uber mode : false
17/07/18 23:05:01 INFO mapreduce.Job: map 0% reduce 0%
17/07/18 23:05:05 INFO mapreduce.Job: map 100% reduce 0%
17/07/18 23:05:11 INFO mapreduce.Job: map 100% reduce 100%
17/07/18 23:05:12 INFO mapreduce.Job: Job job_1500344813707_0002 completed successfully
17/07/18 23:05:12 INFO mapreduce.Job: Counters: 49
```

The above result indicates that the Hadoop installation has been successful.

## View output directory

```
hadoop fs -ls /output
```

## View output result

```
hadoop fs -cat /output/part-r-00000
```

```
[root@VM_96_24_centos /usr/hadoop-2.7.4]# hadoop fs -cat /output/part-r-00000
a          5
dasdada   1
ds         2
qwe        1
ret        1
s          1
v          2
vdfd       1
wqere      1
```

### Note

For detailed procedures on standalone and pseudo-distributed modes, please refer to the official documentation [Getting Started with Hadoop](#).

# COSBrowser

## COSBrowser Overview

Last updated: 2023-09-07 12:06:55

COSBrowser is a visualization interface tool provided by Tencent Cloud COS. You can use it to view, transfer, and manage COS resources easily. Currently, it is available for desktop and mobile clients as described below. If you need to migrate or batch upload data, you can use [Migration Service Platform \(MSP\)](#).

- [User Guide for Desktop Version](#)
- [User Guide for Mobile Edition](#)

### Download Address

COSBrowser	OS	System Requirements	Download Address
Desktop	Windows	Windows 7 32/64-bit or later, Windows Server 2008 R2 64-bit or later	<a href="#">Windows</a>
	macOS	macOS 10.13 or above	<a href="#">macOS</a>
	Linux	Requires a graphical interface and supports <a href="#">Applmage</a> format. Note: To launch the client on CentOS, run <code>./cosbrowser.Applmage --no-sandbox</code> in the terminal.	<a href="#">Linux</a>
Mobile Version	Android	Above Android 4.4	<a href="#">Android</a>
	iOS	iOS 11 or above	<a href="#">iOS</a>
Web	Web	Browsers such as Chrome, Firefox, Safari, and Internet Explorer 10+	<a href="#">Web</a>
Uploader Plugin	Web	Chrome browsers	<a href="#">Offline Download</a>

### COSBrowser Desktop Version

COSBrowser Desktop focuses on resource management, allowing users to upload and download data in bulk.

**Note**

COSBrowser Desktop Version uses a system-configured proxy to connect to the internet. Please make sure that your proxy is set up properly or disable the proxy configuration if it fails to connect to the internet.

- Windows users can find the information in the "Internet Options" of the operating system.
- For queries on macOS, go to **Network Preferences**.
- For queries on Linux, go to **System Settings > Network > Network Proxy**.

COSBrowser Desktop Version has the following features:

SDK	Note
<a href="#">Creating/Deleting buckets</a>	Creates or deletes buckets
<a href="#">Viewing bucket details</a>	Views the basic information of a bucket
<a href="#">Viewing statistics</a>	Views the current storage capacity and number of objects in a bucket
<a href="#">Managing permissions</a>	Modifies the permissions on buckets and objects
<a href="#">Setting versioning</a>	Enables/Suspends bucket versioning
<a href="#">Adding access path</a>	Adds access paths
<a href="#">Uploading files/folders</a>	This feature allows you to upload files/folders to a bucket separately, in batches, or incrementally
<a href="#">Downloading a file/folder</a>	This feature allows you to download files/folders to the local file system separately, in batches, or incrementally
<a href="#">Deleting files/folders</a>	Deletes files/folders from a bucket separately or in batches
<a href="#">Syncing files</a>	Synchronizes local files to a bucket in real time
<a href="#">Copying and pasting files</a>	Copies files/folders separately or in batches from one directory to another
<a href="#">Renaming files</a>	Renames files in a bucket
<a href="#">Creating folders</a>	Creates folders in a bucket
<a href="#">Viewing file details</a>	Views the basic information of files in a bucket
<a href="#">Generating file links</a>	Generates file access links with a certain validity period by requesting a temporary signature
<a href="#">Sharing a file/folder</a>	Shares a file/folder and sets a validity period for the sharing
<a href="#">Exporting file URLs</a>	Exports file URLs in batches

-----	
<a href="#">Previewing a file</a>	Previews media files (images, video, and audio) in your bucket
<a href="#">Searching for files</a>	Searches for files in a bucket through prefix search
<a href="#">Bucket search</a>	Searches for created buckets
<a href="#">Viewing file versions/incomplete multipart uploads</a>	<ul style="list-style-type: none"> <li>• This feature allows you to view the historical versions of files in a bucket where versioning is enabled</li> <li>• View file fragment details within the bucket</li> </ul>
<a href="#">Comparing files</a>	Compares files in a local folder to those in a bucket
<a href="#">Transcoding a video</a>	Transcodes videos with the media processing feature enabled in a bucket
<a href="#">Generating an authorization code</a>	Generates an authorization code for logging in to the COSBrowser client
<a href="#">Processing an image</a>	Scales, crops, or rotates an image, or adds text or image watermarks, and generates a URL of the output image
<a href="#">Setting up a proxy</a>	Sets up a proxy for COS access
<a href="#">Setting the number of concurrent uploads/downloads</a>	Sets the number of concurrent transfers for file upload or download
<a href="#">Setting the number of parts to upload/download</a>	Sets the number of parts for multipart upload or download
<a href="#">Setting the number of retries upon upload/download failure</a>	Sets the number of retries upon upload or download failure
<a href="#">Limiting single-thread upload/download</a>	Limits the upload and download speeds for a single thread

speed	
Setting upload check	Double-checks files uploaded to a bucket
Viewing local logs	Saves records of operations on COSBrowser in the form of local logs

## COSBrowser Mobile Version

COSBrowser Mobile Version is mainly used to monitor COS resources (such as storage usage and traffic) at any time you want. For more supported features, see basic features in [User Guide for Mobile Version](#).

## Changelog

- Desktop Version changelog: [changelog](#).
- Mobile Version changelog: [changelog\\_mobile](#).

## Feedback and Suggestions

If you have any questions or suggestions when using COSBrowser, feel free to give us your feedback:

- Feedback on Desktop Version: [issues](#).
- Feedback on Mobile Version: [issues\\_mobile](#).

## Summary

Of course, COS not only offers the above applications and services but also provides a variety of popular open-source applications integrated with Tencent Cloud COS plugins. Click [here](#) to launch and start using them immediately!

# User Guide for Desktop Version

Last updated: 2024-11-20 19:44:08

## Download and Installation

### Downloading software

OS	System Requirements	Download Address
Windows	Windows 7 32/64-bit or later, Windows Server 2008 R2 64-bit or later	<a href="#">Windows</a>
macOS	macOS 10.13 or above	<a href="#">macOS</a>
Linux	Includes GUI and supports the <a href="#">AppImage</a> format	<a href="#">Linux</a>
Web	Browsers such as Chrome, Firefox, Safari, and Internet Explorer 10+	<a href="#">Web</a>
Uploader Plugin	Chrome browsers	<a href="#">Offline Download</a>

## Installation

You can install the tool by using the installer or decompressing the installation package for your platform. You can also directly use it in browsers without installation.

### Note

To launch the CentOS client, execute the following command in the terminal: `./cosbrowser.AppImage --no-sandbox`.

## Login

You can log in to the COSBrowser desktop version using a permanent key, Tencent Cloud account, or a shared link. You can log in to an account on multiple devices at the same time.

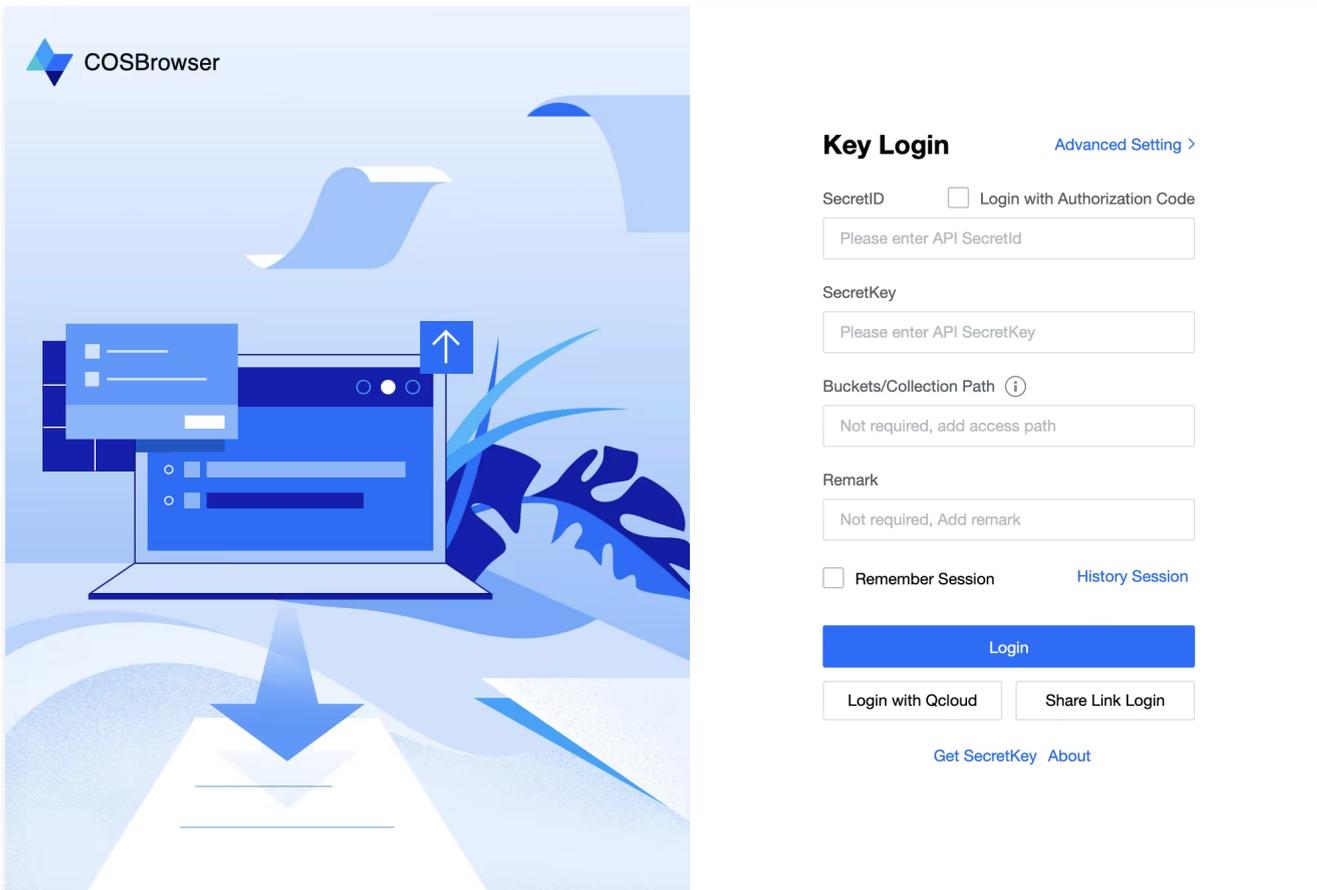
### Login with a permanent key

Log in using Cloud API keys (i.e., SecretID and SecretKey), which can be created and obtained on the [API Key Management](#) page in the Access Management console. After successful login, the keys will be saved in **History Sessions** for easy access next time. The software login interface is shown below. Other configuration item descriptions are as follows:

- **Bucket/Access Path:** If the current key is only allowed to access a specific bucket or a subdirectory within a bucket by the primary account, this field is required. After filling in, you can quickly access the corresponding file path. The format is: Bucket or Bucket/Object-prefix, for example, if the current key is only allowed to access the "doc" folder under the bucket "examplebucket-1250000000", enter "examplebucket-1250000000/doc".
- **Note:** You can provide a description for the current permanent key, such as the operator, purpose, etc. This helps differentiate between different SecretIDs when managing historical sessions in the historical key interface.
- **Remember Session:**
  - If this box is not checked, the Tencent Cloud API key entered will be cleared when you log out (if the key has been saved to the historical sessions, it will be removed).
  - If this box is checked, the Tencent Cloud API key entered will be remembered and can be managed in the historical sessions.

**Note**

COSBrowser does not support logging in with project keys.



**Key Login** [Advanced Setting >](#)

SecretID  Login with Authorization Code  
Please enter API SecretId

SecretKey  
Please enter API SecretKey

Buckets/Collection Path ⓘ  
Not required, add access path

Remark  
Not required, Add remark

Remember Session [History Session](#)

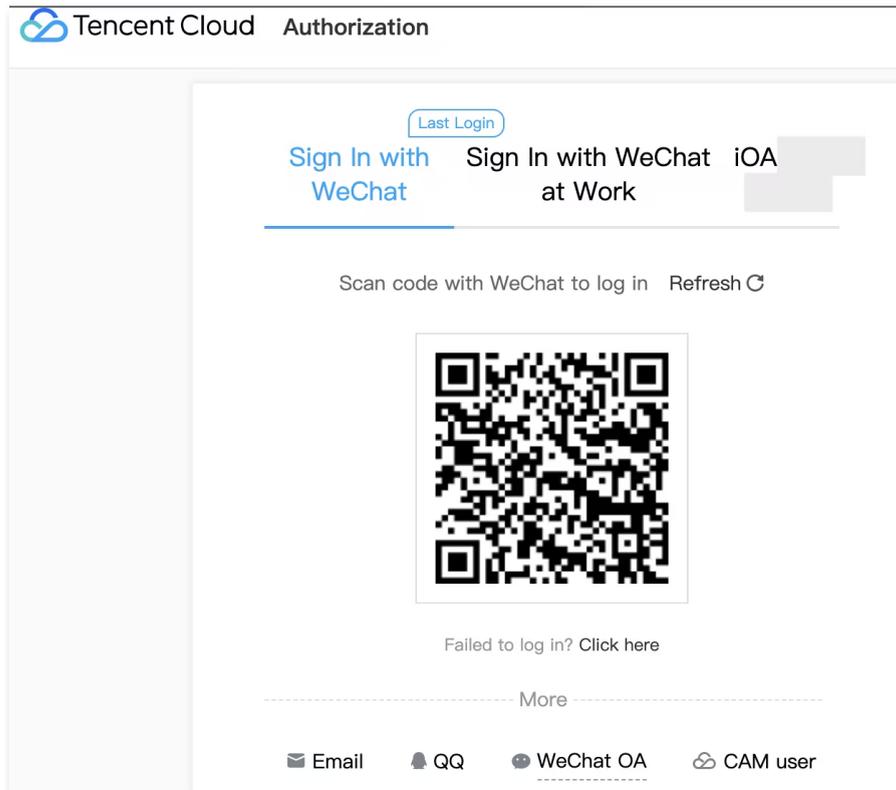
Login

Login with Qcloud Share Link Login

[Get SecretKey](#) [About](#)

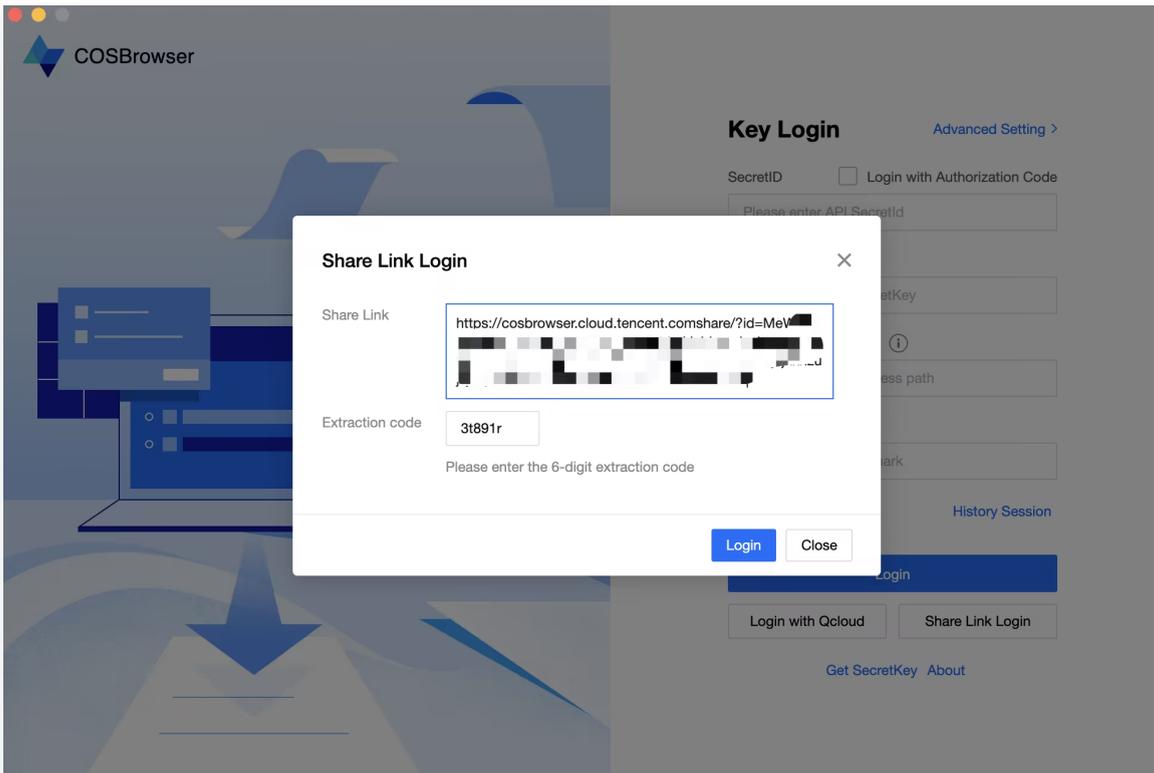
## Login with a Tencent Cloud account

Click **Login with Tencent Cloud Account** and use your Tencent Cloud account to log in to COSBrowser Desktop Version in the pop-up window. You can log in to your Tencent Cloud account via WeChat, email, QQ, WeChat Official Account, WeCom, or sub-user. For detailed directions, see [About Account](#).



## Login with a shared link

You can temporarily log in to the COSBrowser desktop version using a **shared link** and **extraction code** forwarded or shared by others. For a feature introduction, please refer to [Sharing Folders](#).



## Basic Features

### Note

The features introduced here are based on the Windows version v2.11.1. For differences in other versions, please refer to the [Changelog](#).

### 1. Create/Delete Bucket

Feature	Note	Directions
Create a bucket	You can create a bucket directly via the client	<ol style="list-style-type: none"> <li>1. In the Bucket List, click Create Bucket in the top-left corner.</li> <li>2. Enter the correct bucket name, select the region and access permissions, and optionally set bucket tags according to user requirements.</li> <li>3. Click OK to complete the creation.</li> </ol>
Deleting a bucket	Before you delete a bucket, ensure that all data in the bucket has been cleared	<ol style="list-style-type: none"> <li>1. In the bucket list, click the ... &gt; Delete on the right side of the corresponding bucket.</li> <li>2. Confirm that all data in the bucket has been cleared, then click OK to proceed with the deletion.</li> </ol>

## 2. Viewing Bucket Details

You can view bucket details by clicking **Details** on the right of the bucket list. Details include bucket name, region, access permissions, MAZ configuration, versioning status, global acceleration status, and bucket domain name.

### Note

The MAZ configuration currently is only supported in certain regions, such as Beijing, Guangzhou, Shanghai, and Singapore. For more information, see [MAZ Feature Overview](#).

## 3. Viewing Statistics

You can click ... > **Statistics** to view the bucket statistics, including the storage usage and the number of objects.

## 4. Permission Management

You can use COSBrowser to manage permissions for buckets and objects.

- Bucket permissions: Click **Permission Management** on the right of the bucket list.
- Object permissions: Click **Permission Management** on the right of the object.

### Note

For information on COS permissions, refer to [ACL Overview](#).

## 5. Configure Version Control

You can use COSBrowser to enable/disable versioning for a bucket.

Click **Details** in the right-hand operation column of the bucket list. In the bucket details pop-up, click the version control edit icon to perform operations.

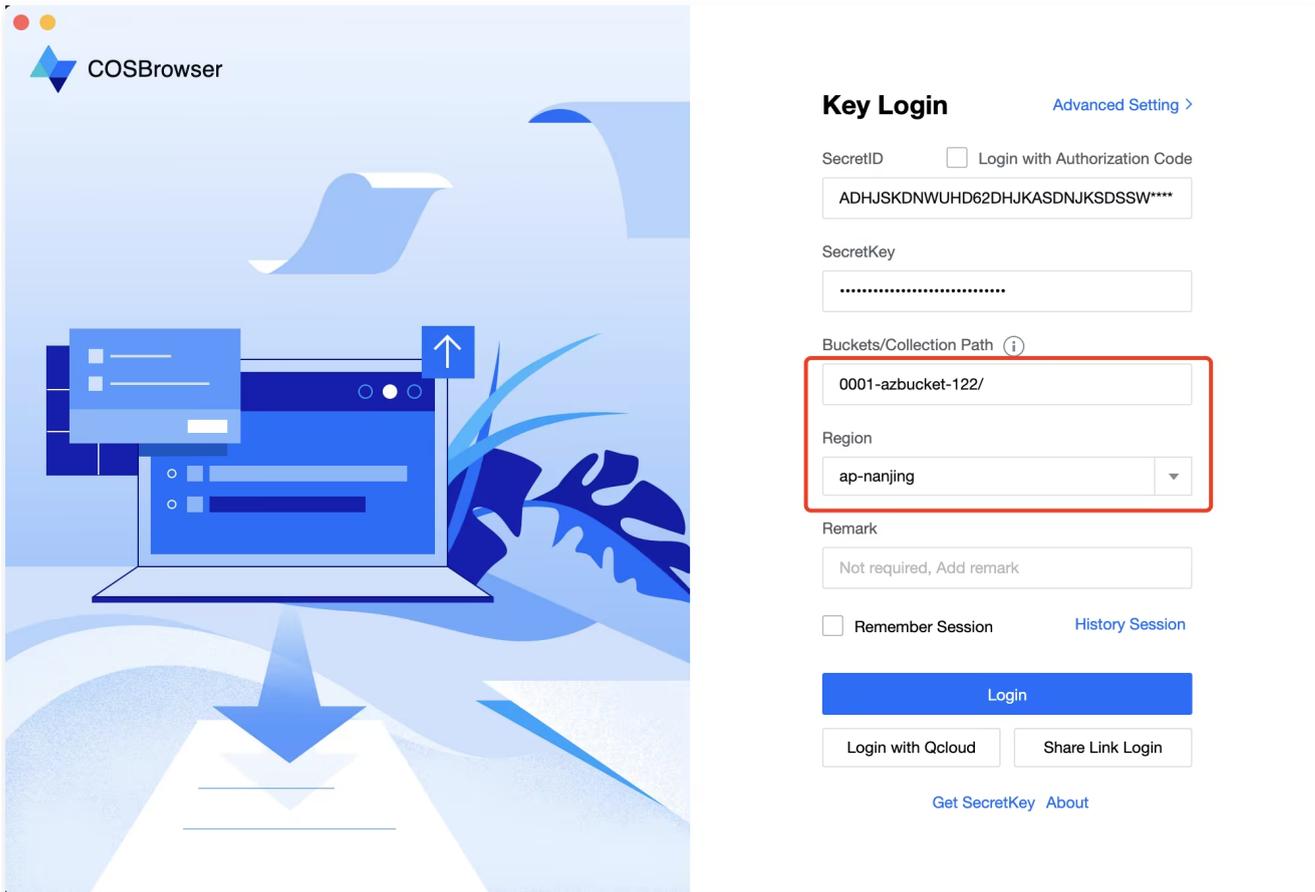
### Note

For more information about versioning, see [Versioning Overview](#).

## 6. Adding Access Path

If you log in with a sub-account that does not have permission to access the bucket list, you can initiate an access via **Add Access Path** in the following two ways:

(1) Add the access path directly on the login screen, select the corresponding bucket region information, and manage resources after logging in.



(2) After logging in with a sub-account, click **Add Path** in the upper left corner of the bucket list page, and enter the specified path to manage resources in the bucket.

## 7. Uploading Files/Folders

Upload Feature	Note	Directions
Uploading a file	COSBrowser allows you to upload a single file or multiple files in batches in different ways.	<p>Several methods for uploading files are as follows, within the specified bucket or path:</p> <ol style="list-style-type: none"> <li>1. Click Upload, select the file, and upload it directly.</li> <li>2. Right-click on an empty space in the file list and select Upload File to begin the upload process.</li> <li>3. Drag and drop files with your mouse into the file list window to upload.</li> </ol>
Uploading a folder and the files contained	If the name of the file/folder to upload already exists in the bucket or path, the old file/folder will be overwritten by default.	<p>The following methods are available for uploading folders to the specified bucket or path:</p> <ol style="list-style-type: none"> <li>1. Click Upload, select a folder, and upload the folder directly.</li> <li>2. Right-click on the blank area in the file list and select "Upload Folder" to upload.</li> <li>3. Drag the folder with your mouse to the file list window to complete the upload.</li> </ol>
Incremental upload	Incremental upload compares the files to upload with existing files in the bucket before the upload. An existing file with the same name in the bucket will be skipped.	<p>The steps for incremental upload are as follows, within the specified bucket or path:</p> <ol style="list-style-type: none"> <li>1. Using the "Upload Folder" method, click "Next" to proceed.</li> <li>2. Skip storage method selection and click Upload to complete the incremental upload.</li> </ol>

### Note

For large-scale file uploads, we recommend a computer configuration of 4 cores and 16GB of RAM (supporting up to 300,000 files in a single upload).

## 8. Downloading Files/Folders

Download Feature	Note	Directions
Downloadi ng a file	COSBrowser allows you to download a single file or multiple files in batches in different ways.	<p>The various methods to download files are as follows:</p> <ol style="list-style-type: none"> <li>1. Select the desired file and click on the download button within the interface to initiate the download.</li> <li>2. Right-click on the selected file and choose Download.</li> <li>3. Download files by dragging them with the mouse to your local computer.</li> </ol>
Downloadi ng a folder and files contained	If the name of the file/folder to download already exists in the local file system, it will be renamed by default.	<p>Several methods for downloading folders and their files are as follows:</p> <ol style="list-style-type: none"> <li>1. Select the desired folder and click the Download button within the interface to initiate the download.</li> <li>2. Right-click and select Download to directly download the folder.</li> <li>3. Drag the folder with your mouse to download it to your local computer.</li> </ol>
Increment al download	Incremental download compares the files to download with local files before the download. An existing file with the same name will be skipped.	<p>Follow the steps below for incremental download:</p> <ol style="list-style-type: none"> <li>1. Right-click on the desired file/folder to download.</li> <li>2. Click Advanced Download in the menu, and select Skip in the pop-up window.</li> <li>3. Click "Download" to complete the incremental download of non-duplicate files/folders.</li> </ol>

 **Note**

If you need to download a large number of files, we recommend a computer configuration with 4 cores and 16GB of RAM (supporting up to 300,000 files in a single download).

## 9. Delete File/Folder

Select the desired file/folder and click **More > Delete** at the top of the page or right-click the file/folder and select **Delete** to delete it. Batch deletion is supported.

## 10. File Synchronization

The file synchronization feature allows you to upload specified files in your local folders to a bucket in real time or at a scheduled time. Detailed directions are as follows:

1. Click **Toolbox > File Sync** in the top-right corner of the page.
2. Configure the following information in the pop-up window.
  - **Local Folder:** Specify the local folder for file synchronization.
  - **Bucket Path:** Specify the destination COS bucket directory.
  - **File Extension Filtering:** Specify the file extensions to be filtered. During synchronization, files with these extensions will be ignored. If there are multiple extensions, separate them with a semicolon `;`.
  - **Sync Type:** Choose between **One-time Sync**, **Auto Sync**, or **Scheduled Sync**, and then click **Start Sync** to initiate the file synchronization feature.

The screenshot shows a 'Sync Setting' dialog box with the following fields and options:

- Local Folder \***: A text input field with a 'Select Local Folder' button and a 'Change' link.
- Bucket Path \***: A dropdown menu labeled 'Select bucket' and a text input field labeled 'Bucket Path'.
- File extension filtering (i)**: A text input field with a placeholder: 'Please enter the file suffix that needs to be filtered. During synchronization, files with this suffix will be ignored. If th'.
- Sync Type**: Three radio buttons: 'Single Sync' (selected), 'Auto Sync', and 'Timing Sync'.
- At the bottom, there is a note: 'Support incremental upload only. See the [documentation](#) for details.'
- Two buttons: 'Start Sync' and 'Clear Local Cache'.

3. Click **Sync Logs** to view the file sync logs.

### Note

- Synchronization means that when the file is uploaded, the system automatically identifies whether the same file exists in the bucket. Only files that do not exist in

the bucket are synchronously uploaded.

- Currently, only synchronizing local files to the bucket is supported. Reverse operation is not supported.
- The file sync feature supports manual, automatic, and scheduled sync.

## 11. Copy and Paste Files

To copy a file/folder, select the desired file/folder in the specified bucket/path and click **Copy** in **More** at the top of the UI. Alternatively, you can right-click it and select **Copy**. After the file/folder is successfully copied, you can paste it to **another bucket or path**. You can copy and paste multiple files/folders in batches.

### Note

For copied files or folders, if the target path for pasting contains files with the same name, they will be overwritten by default.

## 12. File Renaming

Select the file you want to rename, right-click and choose **Rename** or click **Rename** in the **More Actions** menu on the right side of the file. Enter the new filename and confirm to complete the renaming process.

### Note

Folders cannot be renamed.

## 13. Create a Folder

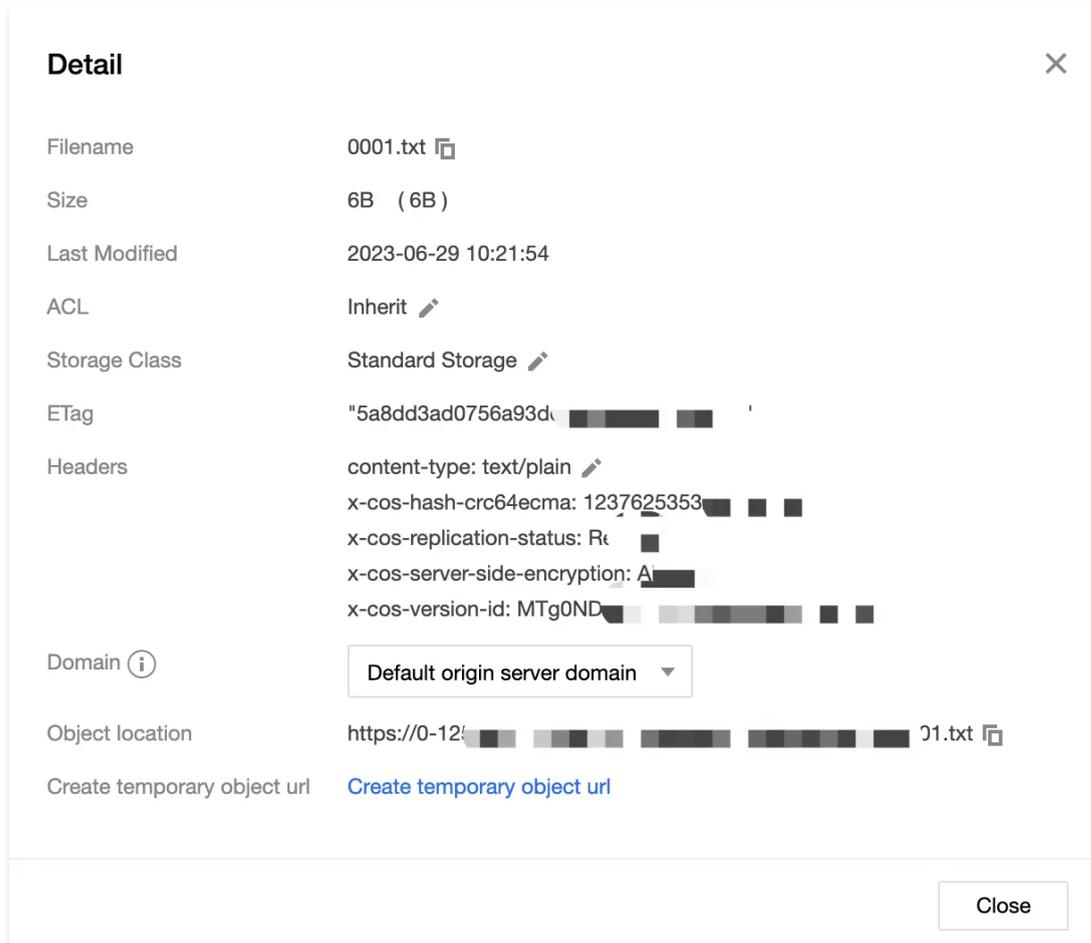
Within the specified bucket or path, click **Create Folder** in the interface or right-click **Create Folder**, enter the folder name and confirm to complete the creation of the folder.

### Note

- The folder name can contain up to 255 characters, including digits, letters, and visible characters.
- Folder names cannot contain special characters such as `\ / : * ? " | < > .`
- The folder name cannot be `..`.
- The folder cannot be renamed.

## 14. View File Details

To view the details of a file, tap its filename or right-click it and select **Detail**. File details include filename, file size, modification time, access permission, storage class, ETag, headers, endpoint, object location, and an option to create a temporary object URL.



## 15. Generate File Link

Each file stored in COS can be accessed through a specific URL. If a file is private-read, you can request a temporary signature to generate a temporary access URL with a certain validity period.

You can generate file links in the following ways:

- In the table view, click the Share icon on the right of the object to generate a URL and copy it. If the file is public-read, the URL will not carry a signature and be valid permanently. If the file is private-read, the URL will carry a signature and be valid for 2 hours.
- Right-click a file and select **Copy Link** to generate a URL and copy it. If the file is public-read, the URL will not carry a signature and be valid permanently. If the file is private-read, the URL will carry a signature and be valid for two hours.
- On the file details page, click **Create link with expires** to set the temporary URL, URL type, and validity period for the specified endpoint (available only when a CDN acceleration endpoint is enabled).

### Copy URL ✕

Filename 0001.txt

Access Private read-write

Domain ⓘ

URL Type  Unsigned URL  
 Signed URL, with validity

The longest valid time is 2 hours

URL `https://0-125[redacted].cloud.com/1/0001.txt?q-sign-algorithm=sha1&q-ak=AKIDFFDjzPXhgAZ[redacted]atmgzw1CUS2L&q-[redacted];1693219[redacted]key-time=1693212033;1693219233&q-header-list=&q-url-param[redacted]e=5bdbf00c7ab0[redacted]bcec4d16f8e170cc921f&x-cos-security-token=e8bf0d3250c0dbf86a13571597ec[redacted].d3320001`

## 16. File/Folder Sharing

Share a folder in COS by clicking the **Share** icon in the toolbar or selecting **Share** from the right-click menu, with the option to set the duration of the share.

### ⓘ Note

- You can only share a single folder but not multiple files.
- If multiple users share a folder, the file content may be hard to manage. In this case, you are advised to enable versioning for your bucket so that you can roll back to the desired historical version.

### Share Folders/Files ✕

Filename 1 ... 1 folders/files.

Permissions  Read only  Read & write

Valid time  hours ▼

The longest valid time is 2 hours

Extraction code  Auto generate  Custom

276

Please enter the 6-digit extraction code

Category	Note
Permission	<p>Sets the access permission for the shared folder.</p> <ul style="list-style-type: none"> <li>• <b>Read-only:</b> Access the link to retrieve the folder list and download files within the folder.</li> <li>• <b>Can edit:</b> Pulls the folder list and downloads files in the folder, upload files to the folder, and create folders using the access URL.</li> </ul>
Validity Period	<p>Units are in minutes, hours, or days.</p> <ul style="list-style-type: none"> <li>• <b>Logging in to the client using a key:</b> The validity period for the primary account ranges from 1 minute to 2 hours, while the validity period for the sub-account ranges from 1 minute to 30 days.</li> <li>• <b>Log in to the client using your Tencent Cloud account:</b> The maximum sharing time that can be set is 2 hours. The default value is the longest validity period allowed for the current account.</li> </ul>
Password	<p>A 6-character password automatically generated by the system. You can customize one as needed (numbers, letters, and symbols are supported).</p>

**Note**

When the link is valid, users receiving the link and password can access the folder.

## 17. Export File Link

COSBrowser supports exporting file links. Click the toolbox icon in the upper right corner of the interface, select **Export File Links** in the toolbox popup, then choose the bucket where the file is located, the folder path to be exported (e.g., to export the folder named "folder" in the root path, enter `folder/`), and the save path. Finally, click **Export** to complete the process.

**Export File Link** ✕

Select bucket

Select Path(Optional)  -

Select Path(Optional)  - +

Save Path  [Change](#)

Export format  CSV

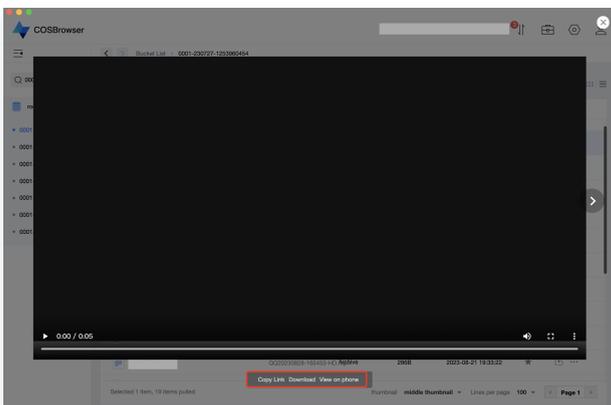
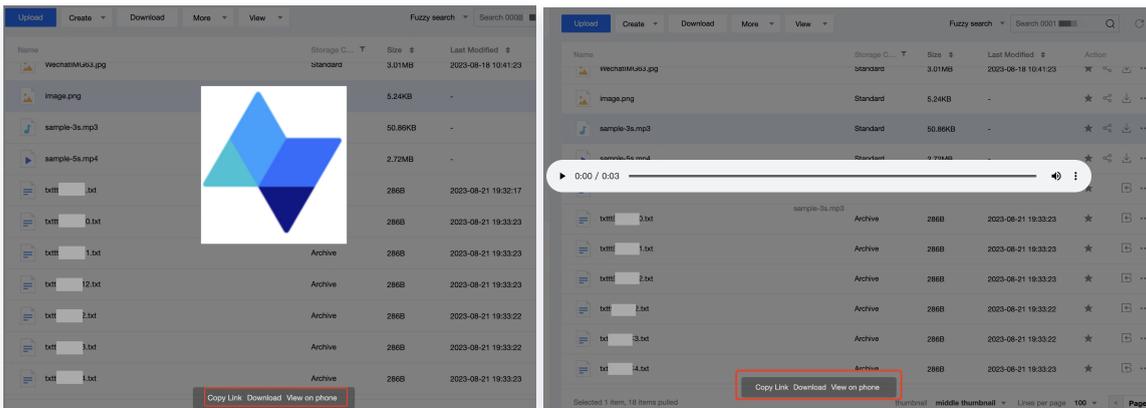
## 18. File Preview

COSBrowser allows you to preview media files, including images, videos, and audio. To preview a media file, double-click it or right-click it and select **Preview** or **Playback** in the context menu. On the file preview or playback screen, you can click:

- **Copy Link** to generate a file access URL and copy it.
- **Download** to download the file to the local file system. If a file with the same name already exists in the local file system, it will be overwritten by default.
- **View on phone** to generate a QR code for the file, which can be scanned on a mobile phone for direct view.

### Note

- Preview is available for images in most formats, .mp4 and .webm videos, and .mp3. and .wav audios.
- Please note that file preview will incur downstream traffic.



## 19. Searching for Files

To search for a file, enter the filename in the search box at the top right of the bucket. COSBrowser supports prefix search and fuzzy search.

## 20. Searching for Buckets

To quickly locate a bucket, enter the bucket name in the search box above the bucket list on the left.

## 21. Viewing Historical Versions or File Fragments

- After enabling version control for your bucket, click **View > Historical Versions** above the file list to see the file's historical versions. In the historical version list, you can search by prefix and clear all historical versions (keeping only the latest version).

### Versions ✕

Clear history version
🔍 Prefix matching

Destination bucket hasn't enabled versioning. If you don't want to overwrite, you can [open](#) versioning.

When versioning is enabled, the historical versions of objects will exist as separate objects. They will occupy storage space and incur storage capacity fees.

Name	Size	Last Modified	VersionId	Action
▼  1.jpeg	75.52KB	2023-08-22 14:46	MTg0NDUwNTEzODY4ODk3OTY1MDY	↓ 🗑️
No history files				
▼  5fbe7ab67d4f30da01fab42a...	Deleted	2023-08-18 11:58	MTg0NDUwNTE3NDI1ODk5MDc0OTQ	🗑️
5fbe7ab67d4f30da01fab42abc031b...	140.96KB	2023-08-18 11:57	MTg0NDUwNTE3NDI2MzMzNDg0ODE	↓ 🗑️
▼  9e85e574d050d47d73eda72f...	184.94KB	2023-08-18 11:57	MTg0NDUwNTE3NDI2MzMzMzA1Mjk	↓ 🗑️
No history files				
▼  ar.docx	638.95KB	2023-08-18 11:57	MTg0NDUwNTE3NDI2MzMxNjE4NzI	↓ 🗑️
No history files				
▼  e8d21e8f58f58810ae0ca715f...	140.29KB	2023-08-18 11:57	MTg0NDUwNTE3NDI2MzMwODAxMTM	↓ 🗑️

Cancel

- When you pause or cancel an object upload, file fragments may be generated. You can click **View > File Fragments** above the file list to view file fragments, with support for searching by prefix and clearing all file fragments.

## 22. File Comparison

In the top-right corner of the UI, click the toolbox icon, select **File Comparison** in the toolbox pop-up window, select the local folder and the bucket (you need to select the region, bucket, and directory) for comparison, and click **Start Comparison**.

### contrast local and bucket folders ✕

Local Folder

Select Local Folder Change

Bucket Path \*

Select bucket ▼

Bucket Path

Region \*

Format such as: ap-guangzhou ▼

Note: The file contrast function currently only supports comparing file names and sizes

Start

Close

## 23. Video Transcoding

In the top-right corner of the UI, click the toolbox icon, select **Video Transcoding** in the toolbox pop-up window, select a bucket for which the media processing service has been enabled, click **Create Transcoding Job**, select the target media file and transcoding template, enter the output file name and storage path, specify the region, bucket, and directory, and click **Transcode**.

Video Transcoding ⓘ 0001-un ✕

Create Task

All ▼

2023-08-28 00:00:00 to 2023-08-28 23:59:59 📅

Please search by task 🔍 ↻

Select Media File

- 0\*0
- 0B
- 00:00:00

↻

H265-MP4- | MP4 | H.265 | 640px \* Proportional height | 400 Kbps ⚙️

Transcoding

Filename Please enter f

Output Path Please select output pa

Selete

⏸ Waiting

## 24. Generate Authorization Key

You can generate authorization codes to temporarily authorize the specified buckets, resources in a bucket, and operations. It is more flexible than folder sharing, as it can grant custom operation permissions for specified directories. You can generate temporary `SecretId`, `SecretKey`, token, and authorization code for other users to log in to the client temporarily. The operations are as detailed below:

In the top-right corner of the UI, click the toolbox icon, select **Generate Authorization Code** in the toolbox pop-up window, select the bucket and scope of resources for authorization, select the authorized operations such as read/write in the policy permission settings, set the authorization code validity period, and click **OK**.

**Generate Authorization Code** [Close]

Buckets \*  54

Resources \*  The whole bucket  Specific resources

Permission Policy Settings \*

effect	Resource	Authorized Action	Condition	Action
allow	0-3az-z-1234567890/*	HeadBucket ListMultipartUploads ListParts GetObject HeadObject OptionsObject	IPequal192.168.1.1	<a href="#">Edit</a> <a href="#">Delete</a>
<a href="#">Add Policy</a>				

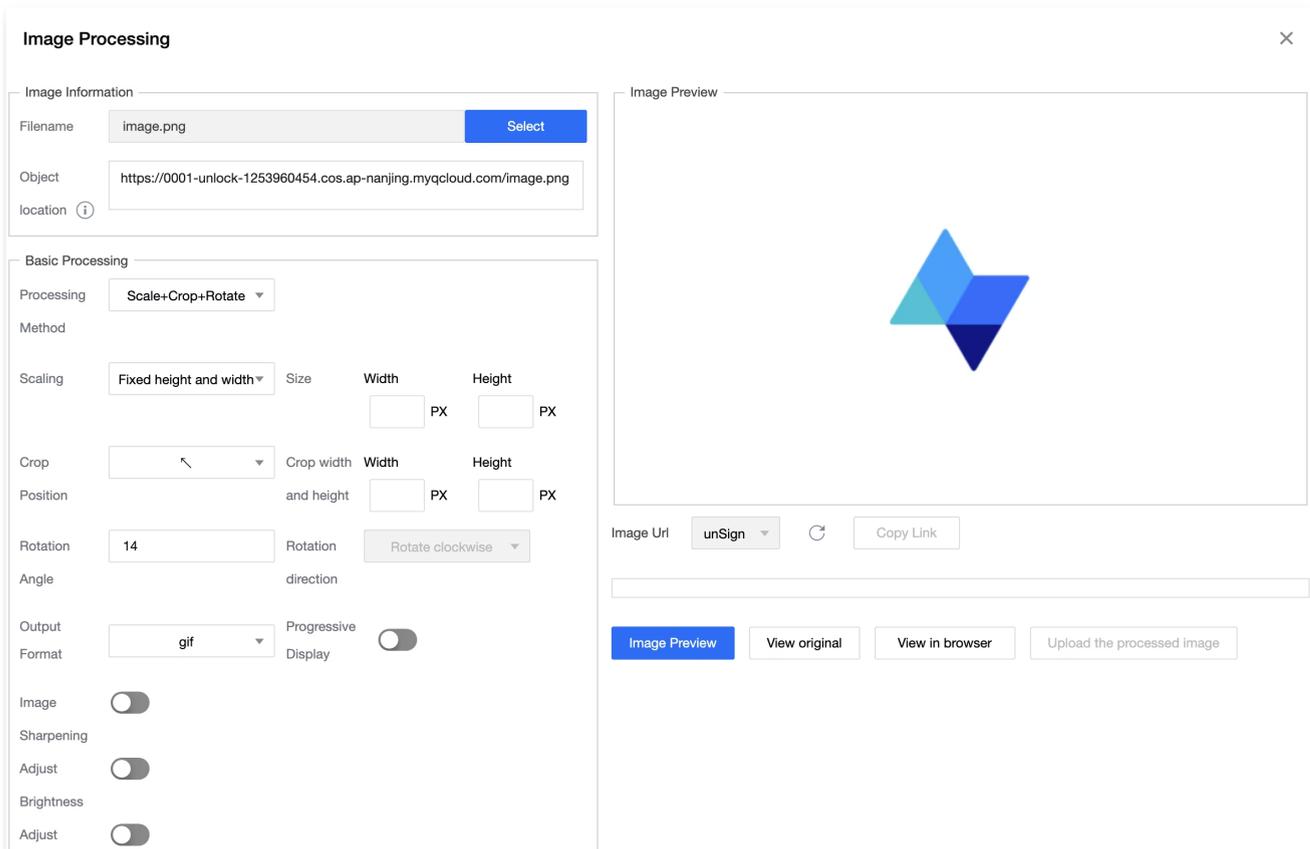
Valid time

The longest valid time is 2 hours

## 25. Image Processing

The image processing feature of COSBrowser supports basic image processing operations, such as scaling, cropping, rotation, and text and image watermarking. It can also generate processed image links.

Select the target bucket, click the toolbox icon in the top-right corner of the UI, and select **Image Processing** in the toolbox pop-up window. In the image processing pop-up window, select the target image file, configure the feature parameters, and click **Image Preview** to generate the link of the output image.



## 26. File List Management

Click the bucket name to access the file list, where you can perform various actions on files or folders, such as uploading files, creating folders, managing file fragments, listing historical versions, searching or refreshing the list, setting folder permissions, viewing file details, previewing, and downloading.

The file list supports two layout styles, which can be toggled in the top-right corner of the file list. Details are as follows:

- **List Layout:** Files and folders are displayed in a list format, which is the default layout.

Upload Create Download More View Fuzzy search Search 0001-unlock- Table layout

Name	Storage C...	Size	Last Modified	Action
1.jpeg	Standard	75.52KB	2023-08-22 14:46:59	★ 🔗 ⏴ ⋮
9e85e574d050d47d73eda72fbc10ff6.jpeg	Standard	184.94KB	2023-08-18 11:57:56	★ 🔗 ⏴ ⋮
QQ20230828-165453-HD.mp4	Standard	96.5KB	-	★ 🔗 ⏴ ⋮
ar.docx	Standard	638.95KB	2023-08-18 11:57:56	★ 🔗 ⏴ ⋮
e8d21e8f58f58810ae0ca715f5b1f2c9.jpeg	Standard	140.29KB	2023-08-18 11:57:56	★ 🔗 ⏴ ⋮
ff32e4de4c0f3f0a8b9e122c49d4a7cc.jpeg	Standard	116.21KB	2023-08-18 11:57:56	★ 🔗 ⏴ ⋮
image.png	Standard	5.24KB	-	★ 🔗 ⏴ ⋮
sss.docx	Standard	9.8KB	2023-08-18 11:57:56	★ 🔗 ⏴ ⋮
u=1535981841,1746583210&fm=253&fmt=auto&app=138&f=JPEG.w...	Standard	23.51KB	2023-08-18 11:57:56	★ 🔗 ⏴ ⋮

- **Grid Layout:** Files and folders are displayed in a grid format, which is not the default layout.

Upload Create Download More View Fuzzy search Search 0001-unlock- Grid layout

Selete All

1.jpeg	9e85e574d050d47d73eda72fbc10ff6.jpeg	QQ20230828-165453-HD.mp4	ar.docx	e8d21e8f58f58810ae0ca715f5b1f2c9.jpeg	ff32e4de4c0f3f0a8b9e122c49d4a7cc.jpeg	image.png	sss.docx
u=1535981841,1746583210&fm=253&fmt=auto&app=138&f=JPEG.w...							

Selected 0 item, 9 items pulled thumbnail middle thumbnail Lines per page 100 Page 1

### Note

In the grid layout, thumbnail functionality is supported, which is technically supported by Tencent Cloud CI. Enabling this feature will incur [Image Processing](#) and [Media Processing](#) fees from Tencent Cloud CI.

## Settings

System Feature	Note	Directions
Setting up proxy	COSBrowser uses the system-configured proxy to connect to the Internet. Please make sure that your proxy is set up properly or disable the proxy configuration if it fails to connect to the internet.	<ol style="list-style-type: none"> <li>1. Select Advanced Settings &gt; Proxy.</li> <li>2. Configure a web proxy for network connections.</li> </ol>
Setting the maximum number of concurrently uploaded/downloaded files	COSBrowser allows you to set the maximum number of concurrently uploaded and downloaded files.	<ol style="list-style-type: none"> <li>1. Navigate to Advanced Settings &gt; Download/Upload.</li> <li>2. Set the concurrency for batch transfers.</li> </ol>
Setting the maximum number of concurrently uploaded/downloaded parts	COSBrowser supports uploading/downloading a file in multiple parts. When the file to be transferred exceeds a certain size, multipart transfer will be performed by default.	<ol style="list-style-type: none"> <li>1. Navigate to Advanced Settings &gt; Download/Upload.</li> <li>2. Set the concurrency for chunked transfers.</li> </ol>
Setting the number of retries upon upload/download failure	COSBrowser will retry failed tasks by default when transferring files.	<ol style="list-style-type: none"> <li>1. Navigate to Advanced Settings &gt; Download/Upload.</li> <li>2. Set the number of retries for failed transfers.</li> </ol>
Setting the single-thread upload/download speed limit	COSBrowser supports limiting the upload and download speeds for a single thread. Total upload (download) speed limit = Single-thread upload (download) speed limit x Number of concurrent files x Number of concurrent parts	<ol style="list-style-type: none"> <li>1. Select Advanced Settings &gt; Upload (or Download).</li> <li>2. Set single-thread upload (or download) speed limit, in MB/s.</li> </ol>
Setting upload check	COSBrowser supports checking files online after upload to verify whether their size and status are correct.	<ol style="list-style-type: none"> <li>1. Navigate to Advanced Settings &gt; Upload.</li> <li>2. Check the box after uploading for secondary verification.</li> </ol>
		<ol style="list-style-type: none"> <li>1. Select Advanced</li> </ol>

## Viewing local log

COSBrowser will record all the performed operations in the `cosbrowsers.log` local file.

- Settings > About.
2. Click on Local Logs, and the system will open the directory where the local logs are located.

# User Guide for Mobile Version Installation and Login

Last updated: 2023-09-07 12:15:13

## Download and Installation

### Downloading software

OS	System Requirements	Download Address
Android	Android 4.4 or Later	<a href="#">Android</a>
iOS	iOS 11 and later	<a href="#">iOS</a>

### Installation

COSBrowser Mobile Version is currently available in most app platforms such as MyApp and App Store. You can download it from the above download address or in an app platform.

## Login Options

COSBrowser Mobile Version supports the following three login options:

- **Login with Weixin:** If your Tencent Cloud account was created through Weixin or associated with a specific Weixin account, you can use the Weixin account to quickly log in to COSBrowser.
- **Login with email:** If your Tencent Cloud account was created through email or associated with a specific email address, you can log in to COSBrowser by entering the email address and password.
- **Login with permanent key:** you can log in using your TencentCloud API key (SecretId and SecretKey; project key is not supported), which can be created or obtained on the [API Key Management](#) page in the CAM console. After successful login, the account will be kept logged in permanently.

#### Note

- If your Tencent Cloud account was created with a QQ account, you can also use the login with Weixin method by selecting login with QQ on the redirected Weixin Mini Program screen.
- If you use a sub-account, you can log in with key or WeChat. For login with WeChat, just select the sub-account on the redirected WeChat Mini Program screen.



# Mobile Version Features

Last updated: 2023-09-07 12:16:15

COS' s COSBrowser mobile app allows you to easily view and manage COS resources anytime you want. The supported operations are described below.

## Data Monitoring

Operation	Description
<a href="#">Usage overview</a>	You can view the recent data usage.
<a href="#">Bucket monitoring</a>	You can view the recent data usage by bucket.
<a href="#">Widget</a>	You can view the data usage without opening the app.

## Bucket Management and Operations

Operation	Description
<a href="#">Viewing bucket list</a>	You can view buckets by region.
<a href="#">Adding access path</a>	If you log in with a sub-account that does not have permission to access the bucket list, you can initiate an access request via <b>Add Access Path</b> .
<a href="#">Creating Bucket</a>	Creates a bucket.
<a href="#">Bucket search</a>	You can fuzzy search for buckets by keyword in the bucket list.
<a href="#">Viewing bucket's basic information</a>	Views the bucket's basic information such as the bucket name, region, creation time, and whether MAZ is enabled.
<a href="#">Bucket permission management</a>	You can modify the public and user permissions of buckets.
<a href="#">Enabling global acceleration</a>	You can enable the global acceleration feature for buckets.
<a href="#">Bucket transfer configuration</a>	You can configure the domain names for uploading and downloading through this feature.

## Object Management and Operations

Operation	Description
<a href="#">Folder creation</a>	You can create folders in buckets.
<a href="#">Folder deletion</a>	You can delete all files in the current directory and all sub-directories.
<a href="#">File upload</a>	You can use COSBrowser Mobile Version to upload local and remote files as well as files from other apps and file managers to COS. You can also set file information such as storage class, access permissions, encryption method, object tags, and metadata during upload.
<a href="#">File backup</a>	COSBrowser provides the automatic backup feature. After you enable backup, it will automatically back up files in your album to the specified bucket.
<a href="#">File download</a>	You can download files in buckets to the app or save them to the local album.
<a href="#">Batch operations</a>	Uploads, downloads, deletes, replicates, or moves objects in the bucket in batches.
<a href="#">Previewing a file</a>	Previews images, videos, audio, documents, and files in other formats in the bucket.
<a href="#">Online file decompression</a>	You can decompress .zip, .tar, and .gz packages online.
<a href="#">File sharing</a>	COSBrowser provides the folder and file sharing feature for you to quickly collect or share data in buckets to other users.
<a href="#">Renaming files</a>	Renames objects in the bucket.
<a href="#">Searching for files</a>	Fuzzy-searches and searching by folder/image/video/audio/documents are supported.
<a href="#">Object sorting or filtering</a>	You can sort files in buckets by filename, size, and modification time and filter them by storage class.
<a href="#">File permission management</a>	You can set file access permissions, which have a higher priority than those for buckets.

## Resource pack management

Operation	Description
<a href="#">Viewing the list of resource packs</a>	You can view the usage of existing resource packs under your current account.
<a href="#">Purchasing resource packs</a>	You can purchase resource packs on a mobile device.
<a href="#">Renewing resource packs</a>	You can renew resource packs on a mobile device.
<a href="#">Upgrading resource packs</a>	You can upgrade resource packs on a mobile device.

# Bucket Management and Operations

Last updated: 2023-09-07 12:16:55

## ! Description

The features discussed here are based on the mobile iOS version 2.7.6. For differences in other versions, please refer to the [Changelog](#).

## Viewing Bucket List

COSBrowser Mobile Version displays bucket lists by region. You can view buckets by region and click **Resources** at the bottom to view created buckets.

## Adding an Access Path

If you log in with a sub-account that has no access to the bucket list, you can add the specified path to a bucket or directory to manage resources by tapping **Add Access Path** in the upper-right corner of the bucket list page.

## ! Note

This feature is available only for sub-accounts.

## Instructions

1. Navigate to the bucket list page and click the "+" icon in the top-right corner.
2. In the pop-up operation list, click **Add Access Path** and enter an access path authorized by the root account.

## Create a bucket.

You can create a bucket by specifying the bucket name, region, and access permissions on COSBrowser Mobile Version.

## Instructions

1. In the bucket list, tap the "+" icon in the upper-right corner.
2. In the pop-up operation list, tap **Create Bucket**.
3. On the bucket creation page, configure the following information:
  - **Name:** a custom bucket name, which cannot be modified once configured. For more information, please see [Naming Conventions](#).

- **Region:** a COS region corresponding to the physical location where your business or users are distributed. This parameter cannot be modified once configured. For more information about regions, please see [Regions and Access Endpoints](#).
  - **Access Permissions:** Three access permissions are available for buckets by default: "Private Read/Write", "Public Read/Private Write", and "Public Read/Write". You can modify it if needed. For more information, please see [Setting Access Permission](#).
4. After verifying the information, click **OK** to create the bucket. In the bucket list interface, you can see the newly created bucket.

## Searching for Bucket

If you have many buckets, you can enter a bucket name in the search box at the top of the bucket list page to fuzzy search for buckets.

## Viewing Bucket's Basic Information

1. On the bucket list page, locate the desired bucket and click the ellipsis (...) on the right.
2. In the pop-up action list, click **Details** to view the basic information of the bucket. The basic information includes the bucket name, region, creation time, and whether Multi-AZ is enabled.

## Managing Bucket Permission

You can use the COS app to set or modify bucket access permissions of the following two types.

- **Public Permissions:** Private Read/Write, Public Read/Private Write, and Public Read/Write. For more information about public permissions, please refer to the [Permission Categories](#) in the Bucket Overview.
  - **User ACLs:** the root account has all bucket permissions (full control) by default. You can add sub-accounts and grant them permissions including read/write, read/write ACL, and even **full control**.
1. Navigate to the bucket list page, locate the bucket for which you need to set or modify access permissions, and click the ellipsis (...) on the right.
  2. In the pop-up operation list, tap **Permission** to enter the bucket permission page.

## Modifying public permission

On the bucket permission page, tap a public permission configuration item to modify it.

## Setting user permissions

On the bucket permission page, tap **Add User** to set bucket access permissions.

## Editing or deleting bucket permission

Select the target user permission, swipe left, and click the displayed **Edit** or **Delete** button to edit or delete the user permission.

## Enabling Global Acceleration

You can enable the global acceleration feature for your bucket on COSBrowser Mobile Version. It allows users across the world to quickly access the bucket and improves your business access success rate and stability. It can accelerate both uploads and downloads.

### Note

Enabling the Global Acceleration feature will not affect the existing default bucket domain name, and you can continue to use it as usual.

## Instructions

1. Navigate to the bucket list page, locate the bucket for which you want to configure global acceleration, and click the ellipsis (...) on the right side of the bucket.
2. In the pop-up operation list, tap **Transfer** to enter the transfer list page.
3. Tap the global acceleration configuration item to enable or disable it as needed.

## Configuring Bucket Transfer

You can enable data transfer via a custom domain name for a bucket on the **Transfer** list page.

### Setting upload domain name

After global acceleration is enabled for a bucket, you can specify the global acceleration endpoint for file upload to a bucket, which will be used first for file uploads once configured.

### Setting download domain name

After a custom endpoint is set for a bucket, you can specify it for file download from a bucket, which will be used first for file downloads once configured.

# File Management and Operations

Last updated: 2023-09-20 10:54:07

## Description

The features discussed here are based on the mobile iOS version 2.7.6. For differences in other versions, please refer to the [Changelog](#).

## Creating folder

COSBrowser Mobile Version allows you to create a folder in a bucket as follows:

1. In the bucket's file list page, click the "+" icon in the top right corner.
2. In the displayed operation list, click **Create Folder**.
3. On the Create Folder page, enter the folder name and click **Confirm** to create the folder successfully.

## Deleting Folder

### Note

Deleting a folder will remove the folder and all files within its subdirectories.

## Instructions

1. Locate the folder you want to delete and click the "..." on the right side of the folder.
2. In the displayed operation list, click **Delete**.

## Uploading a file

You can use COSBrowser Mobile Version to upload local and remote files and files from other apps or the file manager to COS and set information such as file storage class, access permission, encryption method, object tags, and metadata during upload.

## Uploading image/video

COSBrowser allows you to batch upload images or videos from your album to COS.

## Instructions

1. On the bucket list page or file list page, click the "+" icon in the top right corner to display the operation list.
2. In the operation list, click **Upload Images**.
3. In the file list of the displayed album, select the target files and click **Next**.

4. (Optional) Configure the upload parameters. COSBrowser allows you to set the following file attributes during upload:
- **Storage Types:** Based on different business scenarios, you can set different storage types for different objects. The default storage type is STANDARD storage. For an introduction to storage types, see [Storage Types](#).

**Note**

If your bucket has enabled the Multi-AZ configuration, the **Storage Class** can only be set to storage types with Multi-AZ features, such as Multi-AZ STANDARD. If INTELLIGENT TIERING is also enabled, you can choose the Multi-AZ INTELLIGENT TIERING storage class as well.

- **Access Permission:** You can set different access permissions for different objects, with the default permission being inherited (i.e., inheriting the bucket's permission). For more information about access permissions, see [Access Control Basic Concepts](#).
- **Server-Side Encryption:** configure server-side encryption for the object you want to upload. COS will automatically encrypt your data as it is written and decrypt it when you access it. Currently, COS offers two encryption types: SSE-KMS (only available in Beijing, Shanghai, and Guangzhou regions) and SSE-COS. For more information, please see [Server-side Encryption Overview](#).
- **Object Tags:** Object tags consist of a tag key (tagKey), tag value (tagValue), and an equal sign (=), for example, group = IT. You can set, query, and delete tags for specified objects.
- **Metadata:** object metadata, or HTTP header, is a string sent by the server over HTTP before it sends HTML data to the browser. By modifying HTTP headers, you can modify how the webpage responds as well as certain configurations, such as caching time. Modifying an object's HTTP headers does not modify the object itself. For more information, see [Custom Headers](#).

5. Click **Upload**.

## Uploading file through link

COSBrowser allows you to upload a file through the file link. Every time you enter the app, it will check the current clipboard. If there is any valid file link on the clipboard, a message will pop up asking you whether to upload the file through the link. In this case, you simply need to click **Upload Now**.

You can also upload a file through the link as follows:

1. On the bucket list page or file list page, click the "+" icon in the top right corner to display the operation list.

2. Click **Upload Link**, paste the link for file upload into the text box, select an upload path, and click **Upload**.

## Uploading file shared by third-party app

You can also share a file from another app to COSBrowser to upload it.

### **Note**

This feature requires third-party apps to support sharing files with other apps.

## Instructions

Take QQ as an example:

1. In the QQ App, first click on the file to preview it, then click on the "..." in the top right corner, and **select Other Apps**.
2. Click COSBrowser in the app list.

## Uploading file from file manager

COSBrowser can upload files from the system file manager ("Files" on iOS and the corresponding system file manager app on Android) to COS.

## Instructions

1. Navigate to the bucket or file list page, and click the "+" icon.
2. In the displayed operation list, click **Upload Files**.
3. On the displayed file manager page, click the target file.

## Backing up objects

COSBrowser provides the auto backup feature. After this feature is enabled, COSBrowser will automatically back up the files in your album to the specified bucket. To help you manage backup files, COSBrowser displays the backup data as a separate module on the homepage.

### **Note**

- The album backup feature is only supported for the root account.
- The album module displays only image and video files. To view all files, go to the bucket list and search for the backup bucket.

## Setting backup

Navigate to **Profile > Photo Backup** and click to enable automatic backup of photos or videos:

- **Automatic Photo Backup:** After it is enabled, all images in the album will be backed up.

- **Automatic Video Backup:** After it is enabled, all videos in the album will be backed up.
- **Back up over Wi-Fi Only:** After it is enabled, files will be backed up only over a Wi-Fi network.
- **Region:** After selecting the backup region, we will create a default bucket in that region with the name format "from-phone-date-APPID".

#### Note

The associated region can only be set before enabling backup and cannot be modified after saving.

- **Enable Smart Storage:** This option is suitable for buckets with INTELLIGENT TIERING enabled. After it is enabled, files in the album will be uploaded to COS in INTELLIGENT TIERING. COS will automatically switch between the STANDARD and STANDARD\_IA classes based on the access frequency of INTELLIGENT TIERING objects with no data retrieval fees incurred, which reduces your storage costs. For more information, see [INTELLIGENT TIERING Overview](#).

## Managing backup file

COSBrowser supports batch file upload and download.

### Instructions

1. On the **Album** page, click the batch operation icon in the top-right corner to display operation buttons.
2. Select the target files and click **Download** or **Delete** to download or delete the files.

## Adding file to backup bucket

You can also add image or video files from other buckets to the backup bucket to quickly preview them in the **Album** module.

### Instructions

1. Locate the image or video file you want to add, and click the "..." on the right side.
2. In the displayed operation list, click **Add to Album**.
3. Enter the **Album** module again to view the file.

## Downloading objects

COSBrowser allows you to download files from COS to your local file system. It also enables you to save images to the system album.

## Downloading to app

COSBrowser provides multiple download entries for you to download files anytime, anywhere.

- Method 1:

1. Navigate to the file list page, locate the file you want to download, and click the "..." on the right side.
2. In the displayed operation list, click **Download**.

- Method 2:

Enter the file details page, click the "..." in the top right corner, **a list of operations will appear, click Download to proceed.**

- Method 2:

Enter the file preview page, click the "..." in the top right corner, **a list of operations will appear, click Download to proceed.**

## Saving to album

COSBrowser allows you to save images to the local album as follows:

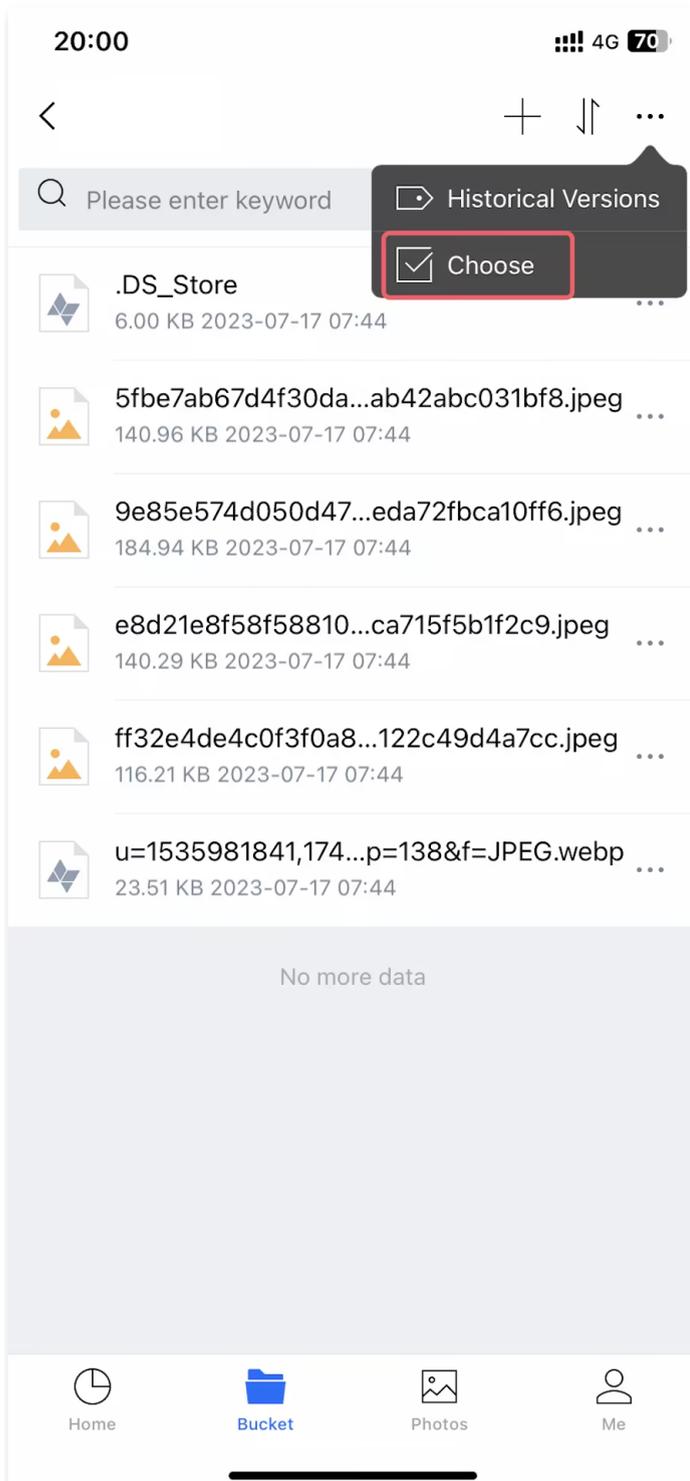
1. Navigate to the file list page, locate the file you want to download, and click the "..." on the right side to display the operation list.
2. Click **Download** and select **Save to Album**.

## Operating in batches

COSBrowser allows you to batch download, delete, copy, and move files from a bucket.

### Batch download

1. Click on the bucket to enter the file list page, and click the "..." icon in the top right corner.
2. In the pop-up menu, click **Batch Operation**.

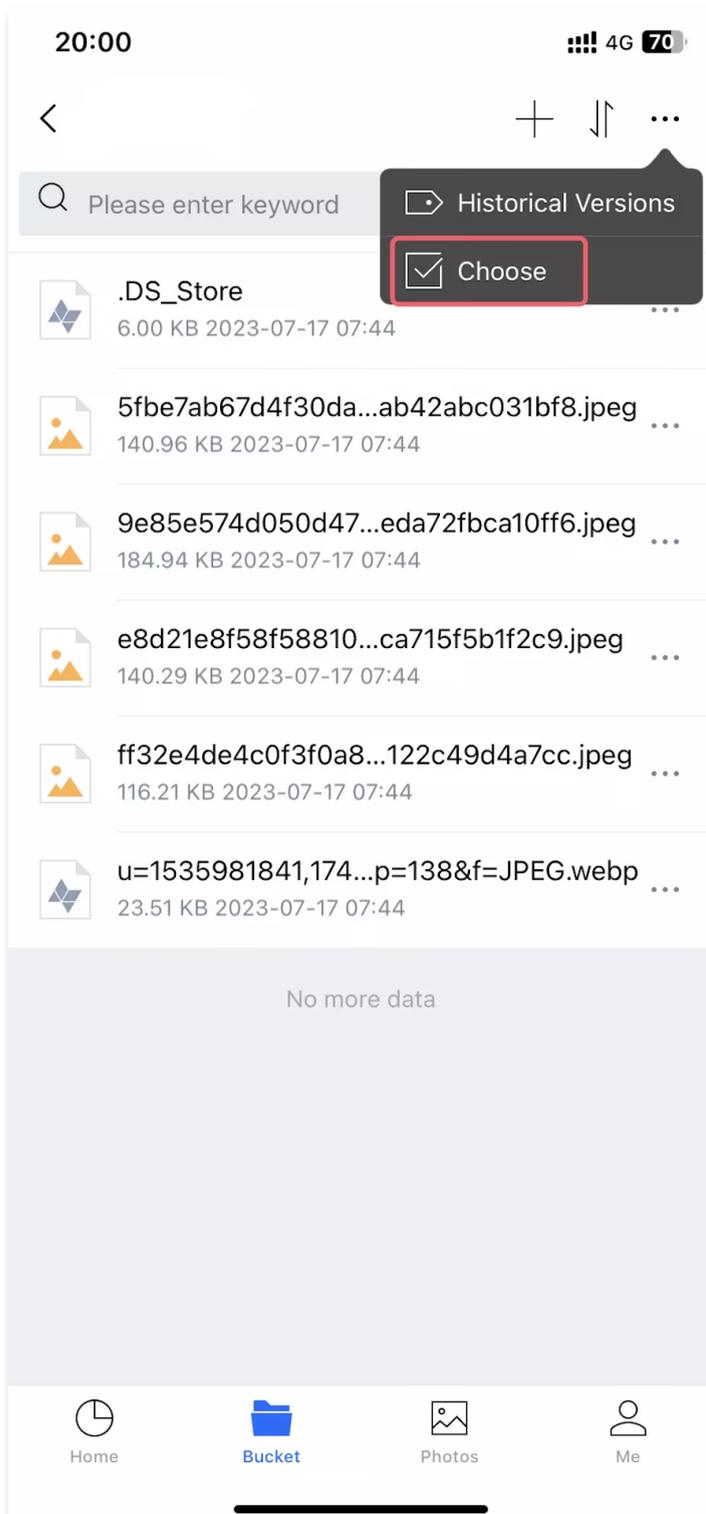


3. Select multiple files and click **Download** in the bottom action bar to download the selected files in bulk to the app.

You can also click the transfer button in the top right corner of the file list page to view related tasks in the transfer list page.

## Deleting domain names by batches

1. Click on the bucket to enter the file list page, and click the "..." icon in the top right corner.
2. In the pop-up menu, click **Batch Operation**.



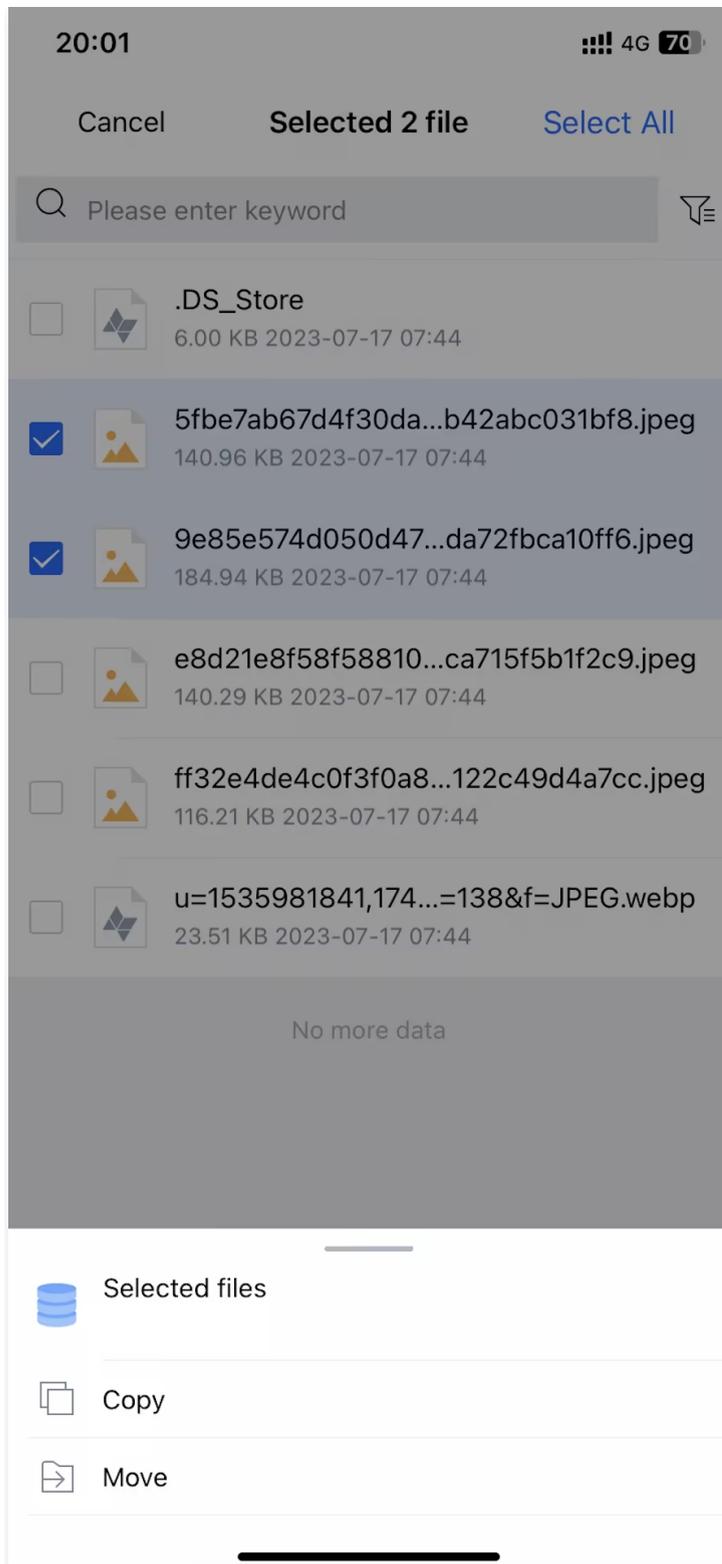
3. Batch select files and click **Delete** on the bottom operation bar.

## Batch copy

### Note

During the copying process, COSBrowser will also copy the source file's ACL, Policy, Tagging, and metadata along with other related information.

1. Click on the bucket to enter the file list page, and click the "..." icon in the top right corner.
2. In the pop-up menu, click **Batch Operation**.



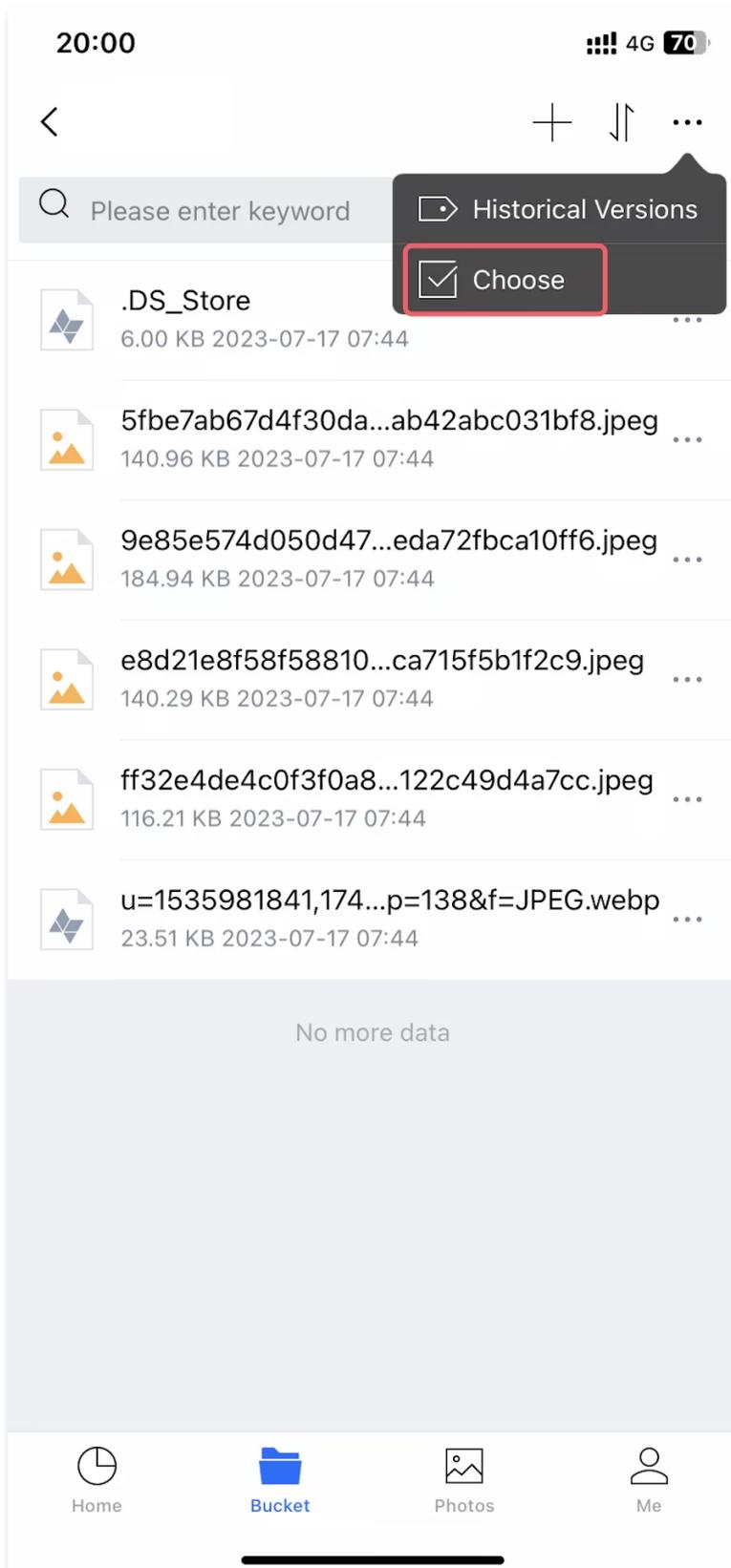
3. Select multiple files and click **More > Copy** in the bottom action bar to batch copy the selected files to a new bucket or folder.

## Batch moving

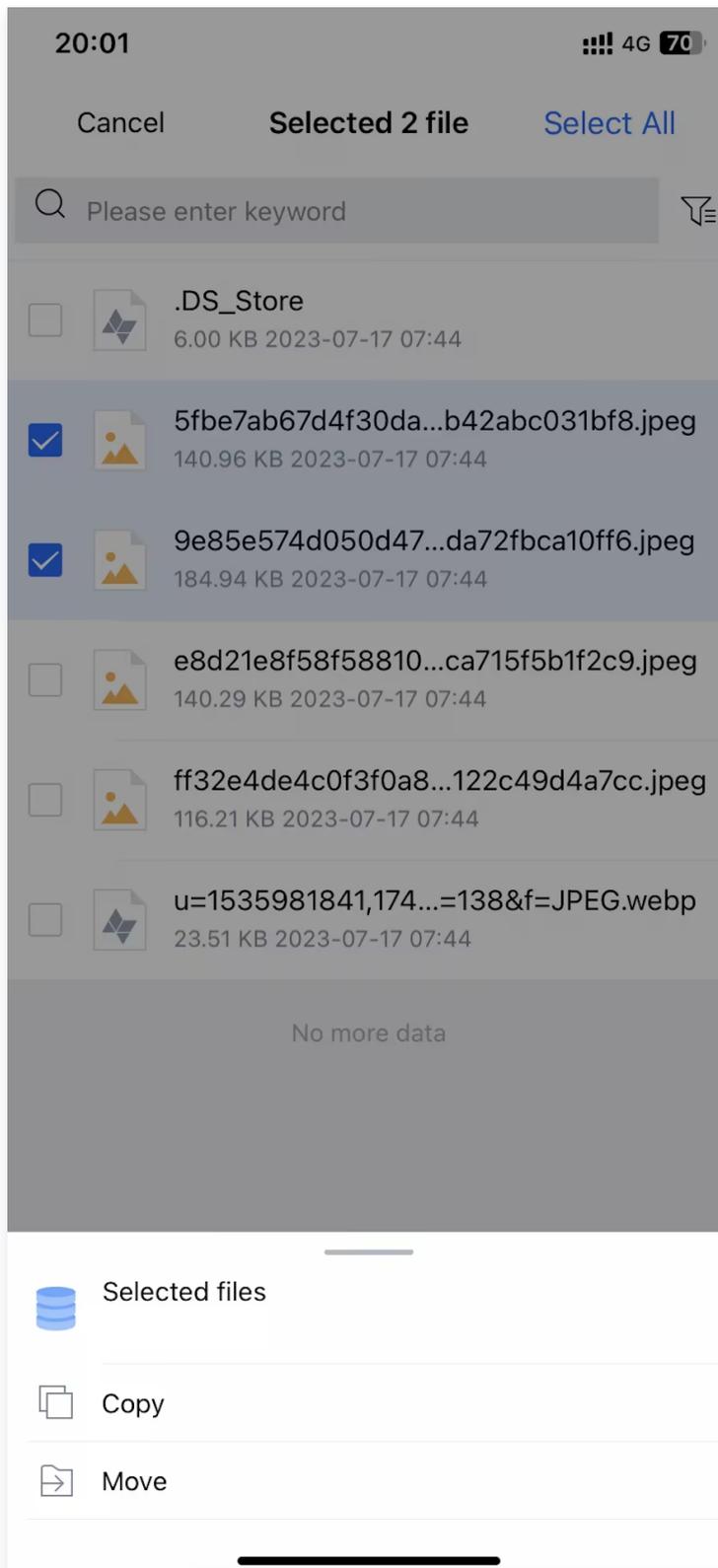
### Note

During the migration process, COSBrowser will also transfer the source file's ACL, Policy, Tagging, and metadata along with other related information.

1. Click on the bucket to enter the file list page, and click the "..." icon in the top right corner.
2. In the pop-up menu, click **Batch Operation**.



3. Select multiple files and click **More > Move** in the bottom operation bar to move the selected files in bulk to a new bucket or folder.



## Previewing file

COSBrowser allows you to preview various file formats such as image, audio, video, and document. It also supports online file decompression.

## Online image preview

COSBrowser allows you to preview images online without downloading them.

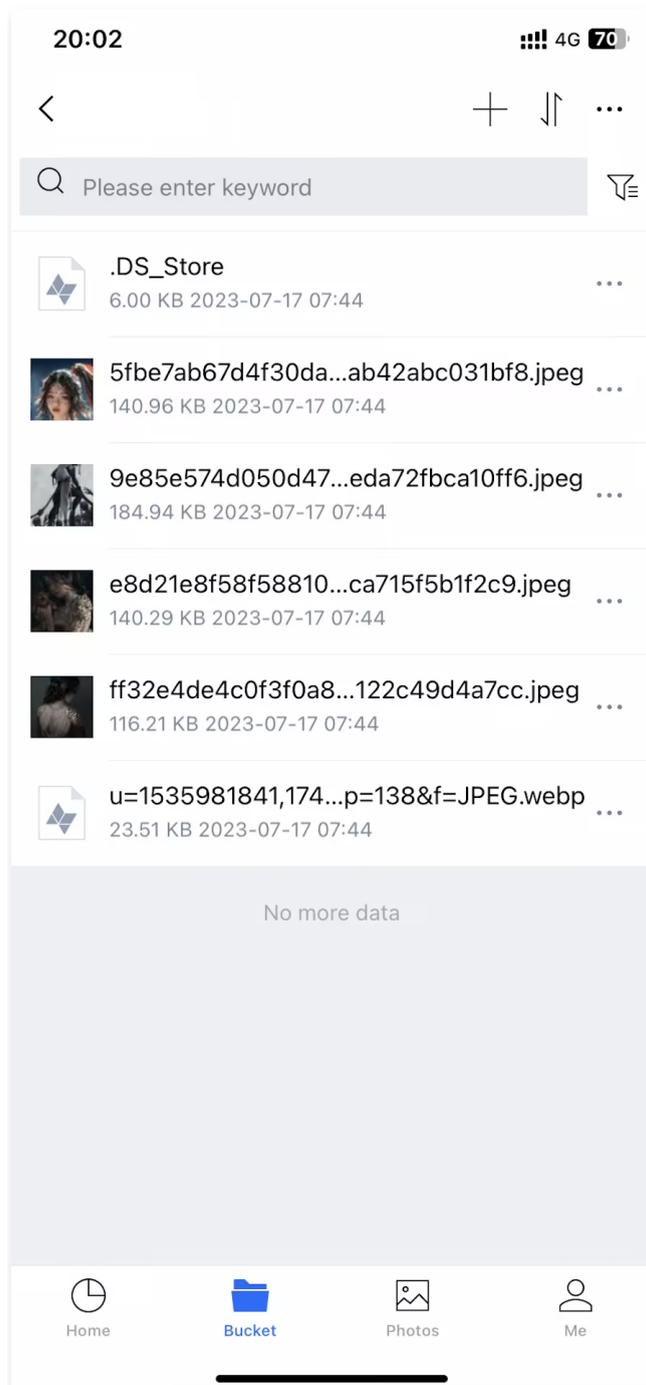
## Enabling image preview

You can preview images in the following three methods:

### 1. Enable the feature globally

Navigate to **Profile > Settings** and toggle the switch to enable the **Preview Image** feature globally.

The effect after enabling is shown in the figure :



### 2. Enable the feature temporarily

Upon first launching the App and entering the file list page, the App will check if there are any images in the current list. If there are, a prompt to enable image preview will appear. You can click **Enable Preview** to quickly activate the **Preview Image** feature; if not needed, click **Cancel** to ignore. If you wish to use this feature later, you can manually enable it in **Profile > Settings**.

### 3. Enable the feature for one image

If you have disabled image preview in **Personal > Settings** and it's not your first time opening the app, you will be prompted to enable preview when entering the image details. Click **Click to Preview** to preview a single image.

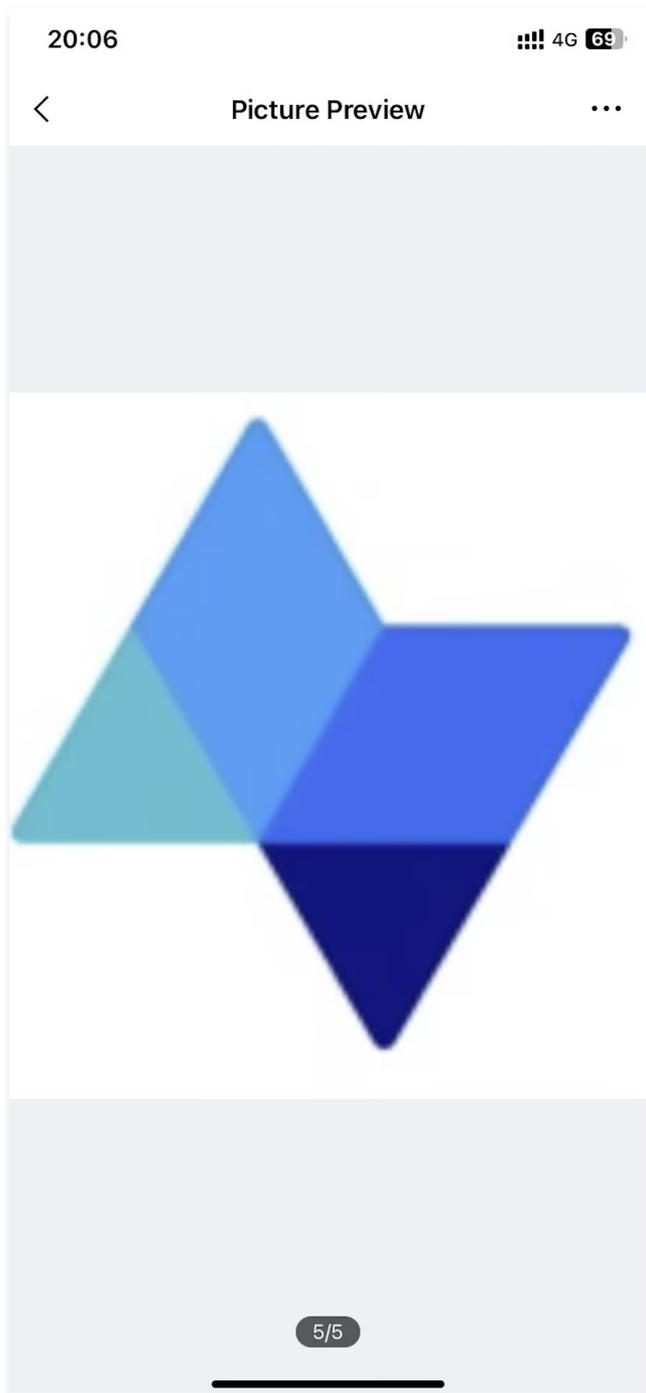
#### **Note**

This toggle will not change the status of the global **Image Preview** switch.

## Big image mode

COSBrowser provides the big image mode feature for you to view image details more clearly.

1. On the file list page, click the target image to enter the **File Details** page.
2. Click on the image to enter the full-screen viewing mode, where you can use two fingers to zoom in and view the details of the image.

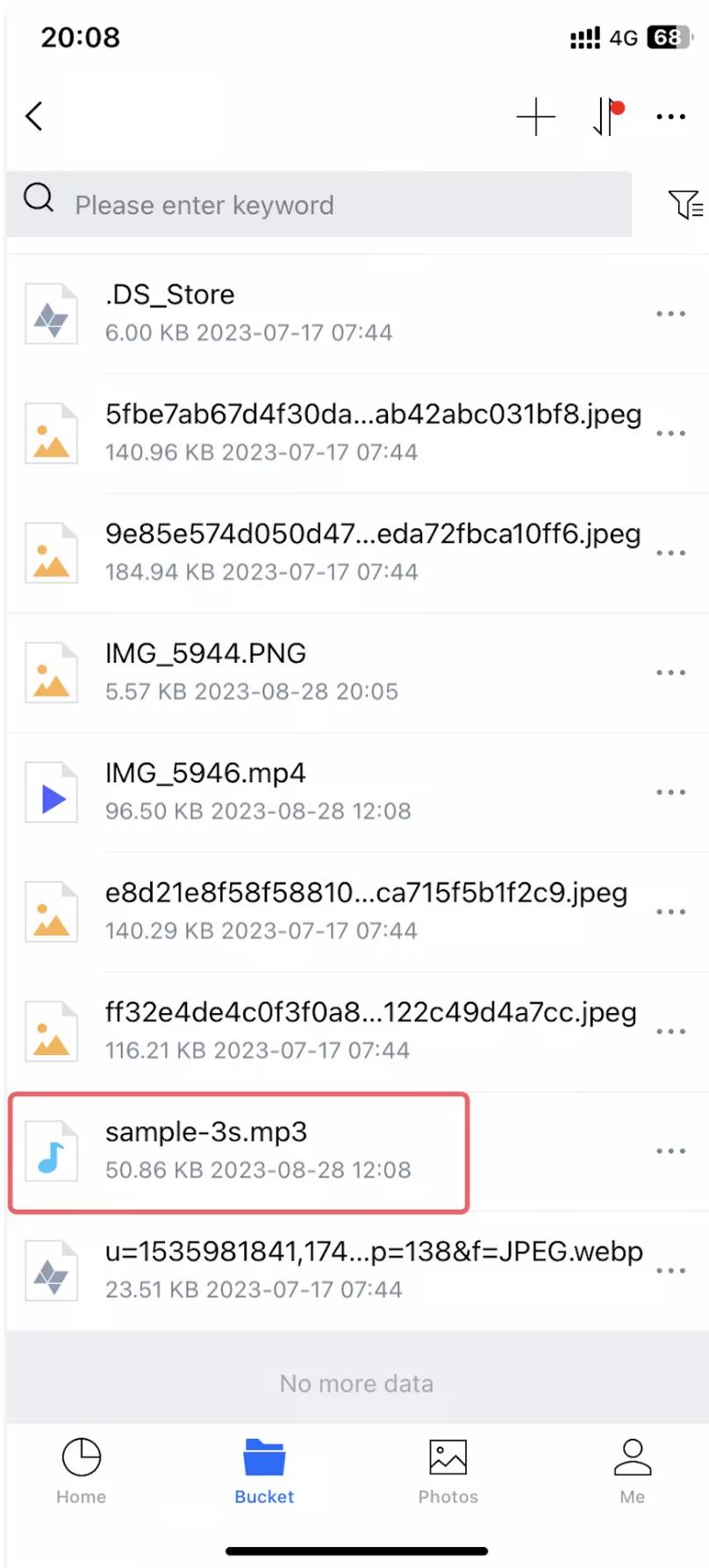


## Playing back audio

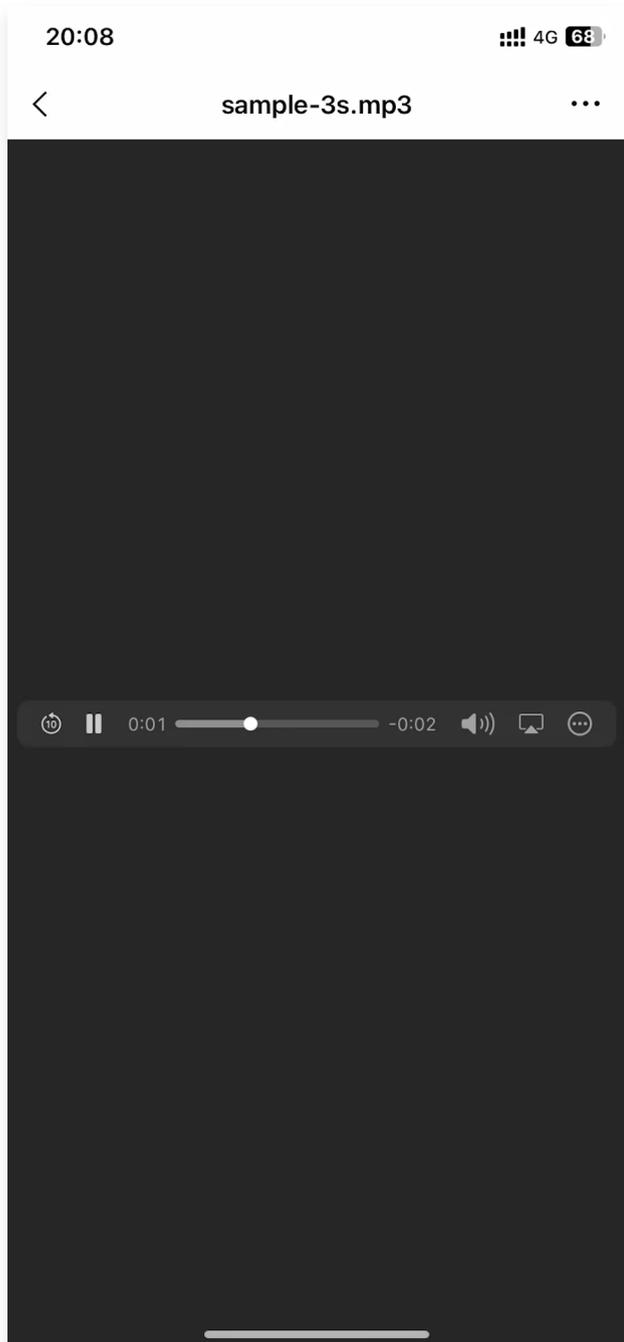
COSBrowser can play back audio files in MP3, OGG, AAC, WMA, WAV, APE, or FLAG format.

### Instructions

1. Navigate to the file list page, click on the audio file to enter the file details.



2. Click the play button to start playback.  
During playback, you can fast forward and pause.

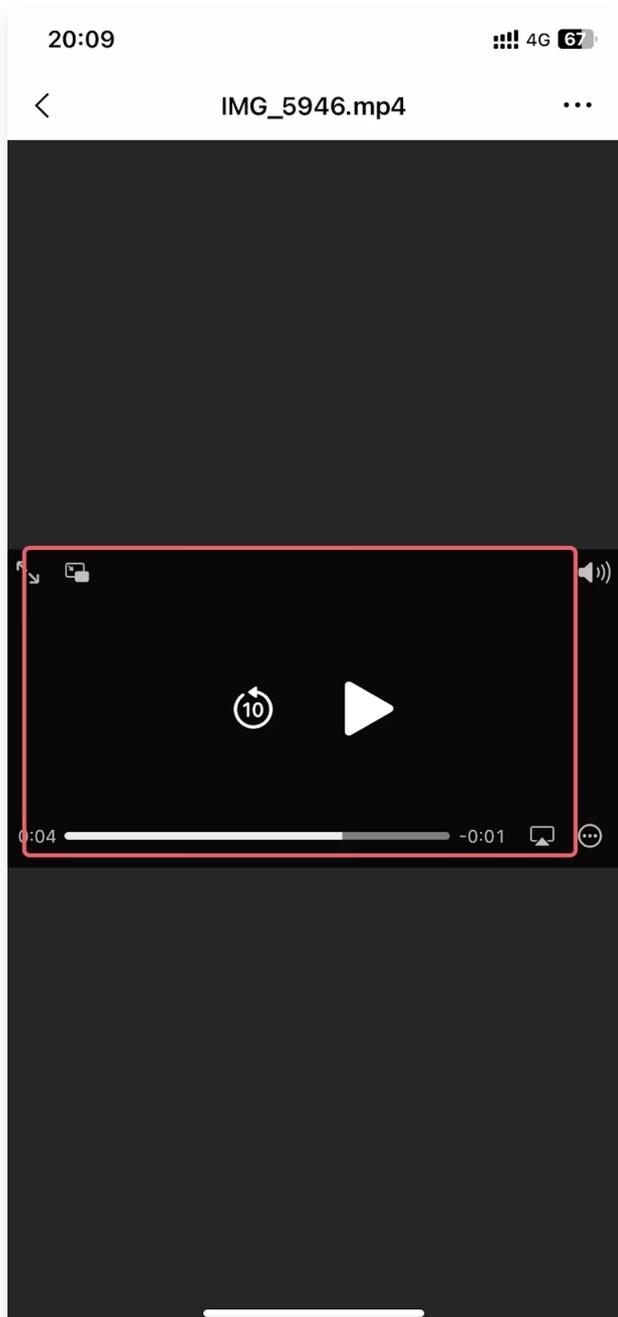


## Playing back video

COSBrowser provides a simple online video playback feature, which supports various formats such as AVI, WMV, MPEG, RM, RMVB, MKV, MOV, QT, and MP4. You can speed up or pause the video playback during watch. You can also play back the video in a floating window after the app enters the background.

## Instructions

1. On the file list page, click the target video file to enter the **File Details** page.
2. Click Play to start playback.  
During playback, you can fast-forward and pause.

**Note**

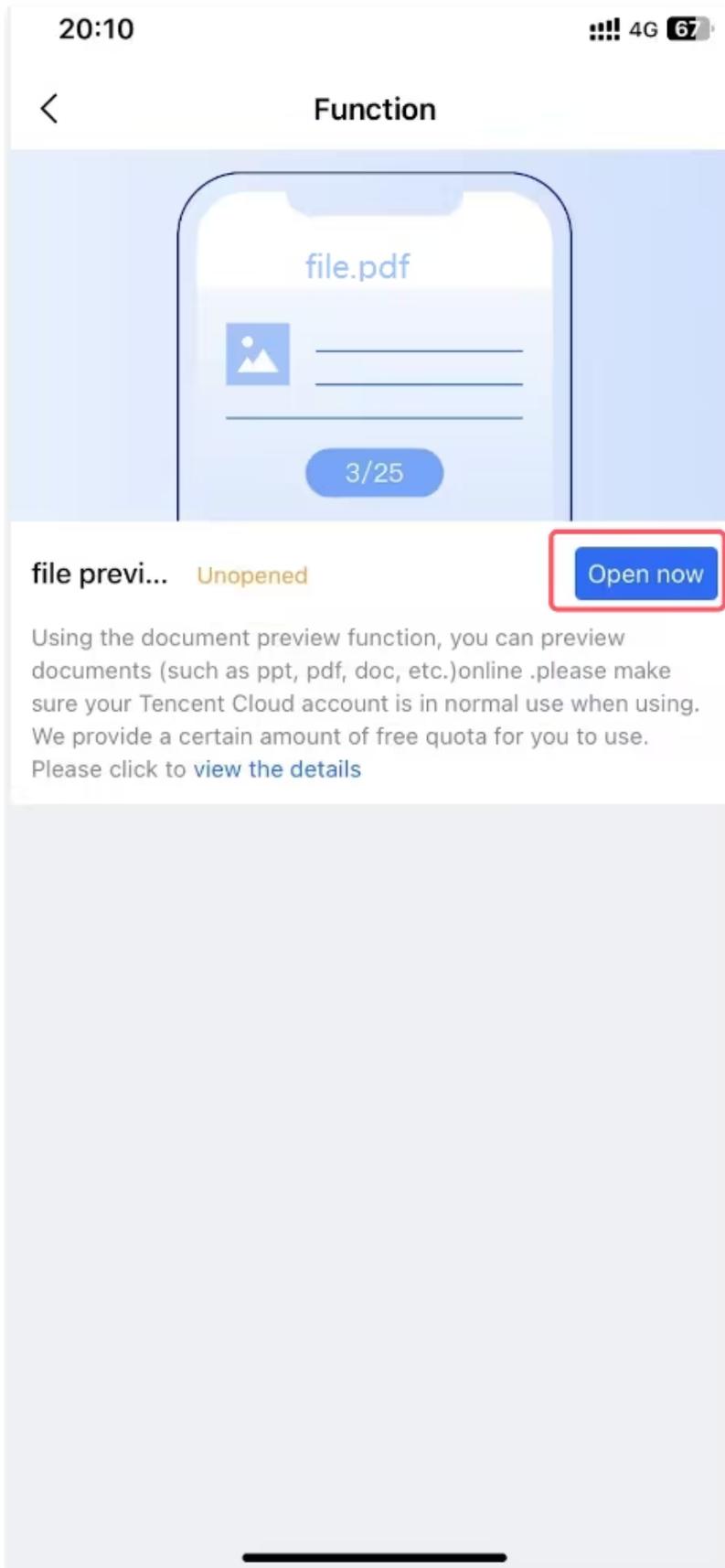
When the app enters the background, you can still watch videos using a floating window.

## Previewing document

COSBrowser allows you to preview files in multiple formats such as PDF.

## Instructions

1. Click on the document you want to preview to enter the file details page. If the document preview feature is not yet enabled, click **Enable**.
2. Navigate to the document preview configuration page and click **Enable Now** to activate the document preview feature.



3. After successfully enabling, return to the previous page and click **Click to Preview**.

20:11

4G 67



### File Detail



Basic Info

Permissions Info



Document Preview is enabled

[Click Preview](#)

Object Name	txttt   :1.txt
Object Size	286 bytes
Modify time	2023-08-02 18:36
Encryption	SSE-COS
Storage Type	STANDARD

Designated Domain	Default COS Domain
Object Link	https:// 0003-230717-1317301984.cos.a p-beijing.myqcloud.com/ txttt%E7%9A%84%E5%89%AF %E6%9C%AC1.txt
ValidityTime	1 小时
Object Tag	0
Metadata	1 >

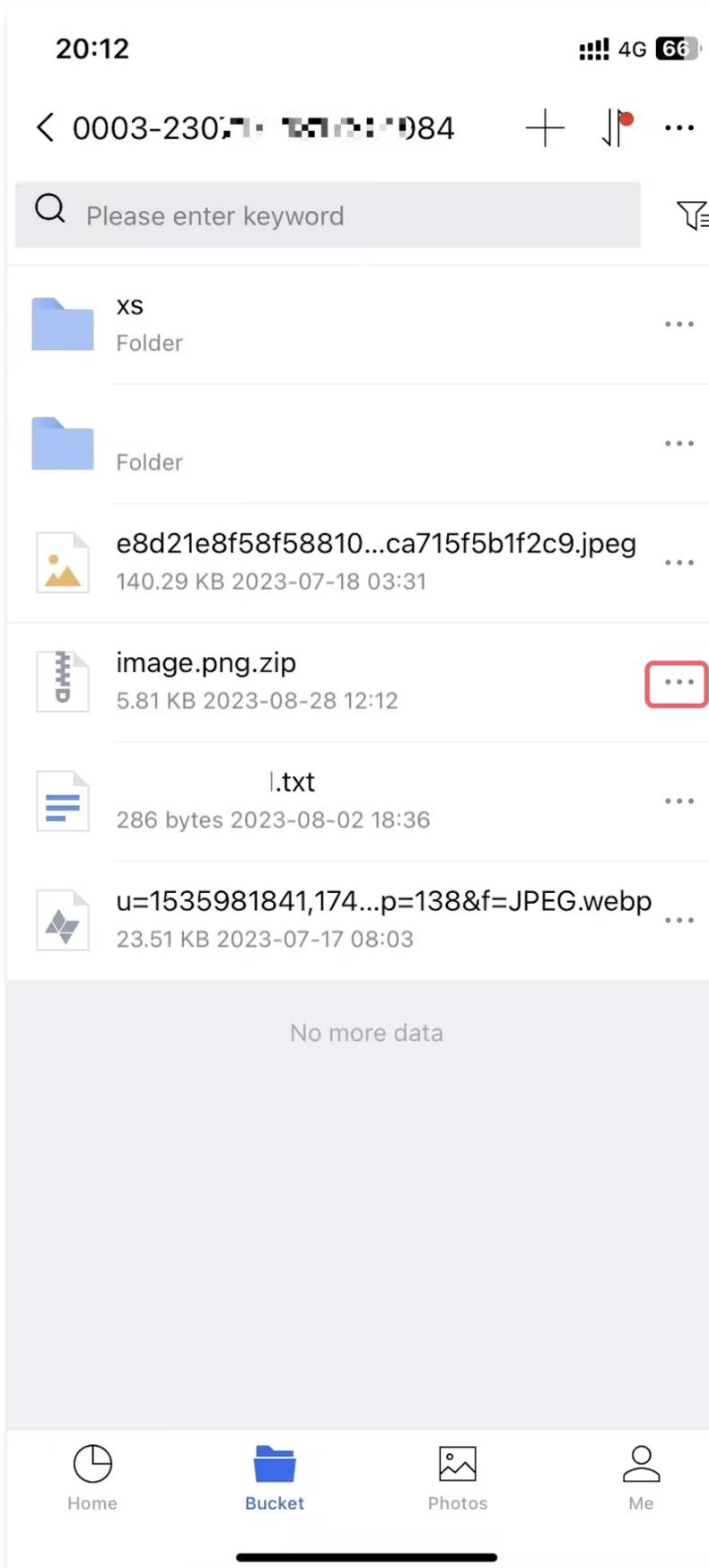
[Create share link](#)

4. Navigate to the document preview page, swipe left to view the next page.

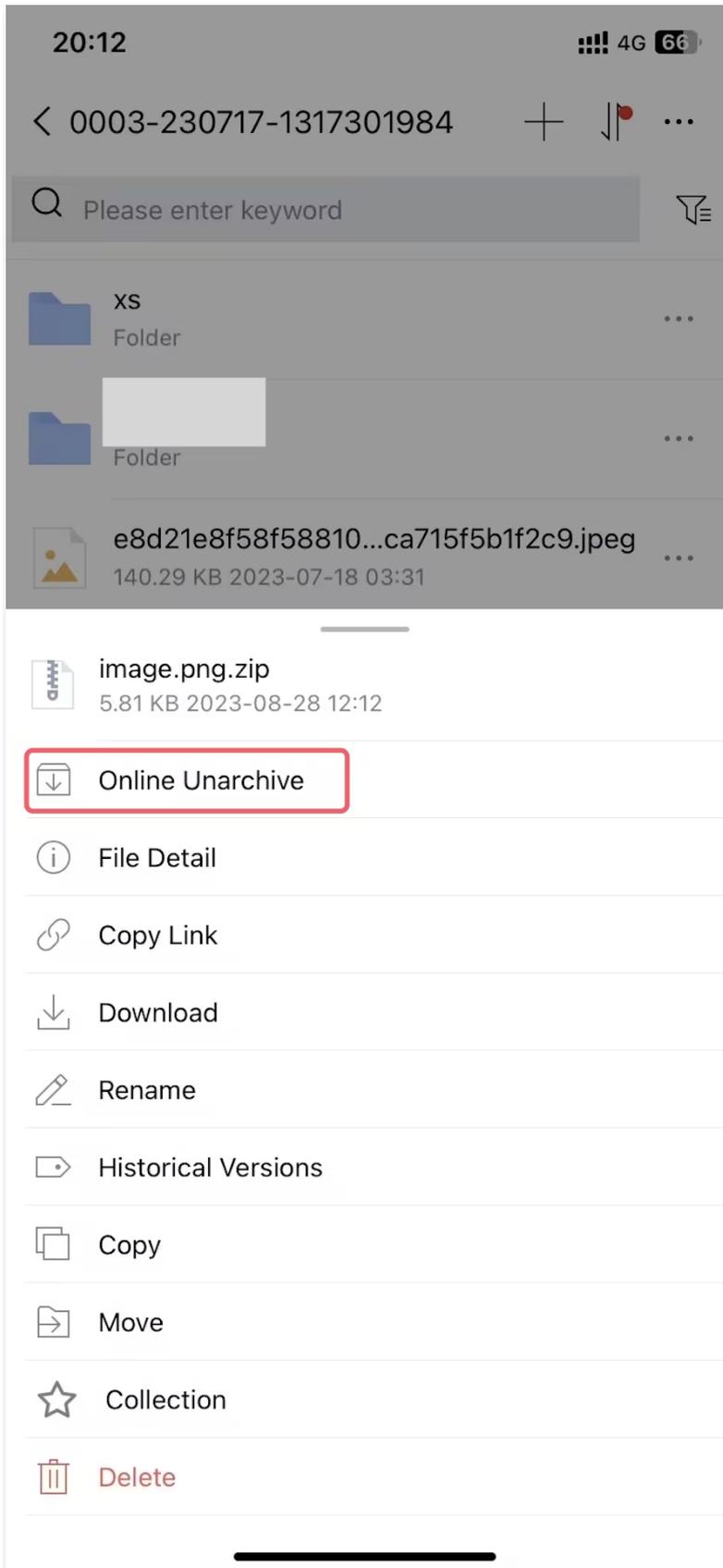
## Decompressing Files

COSBrowser allows you to decompress files in ZIP, TAR, or GZ format online. The extracted files will be stored in the current directory.

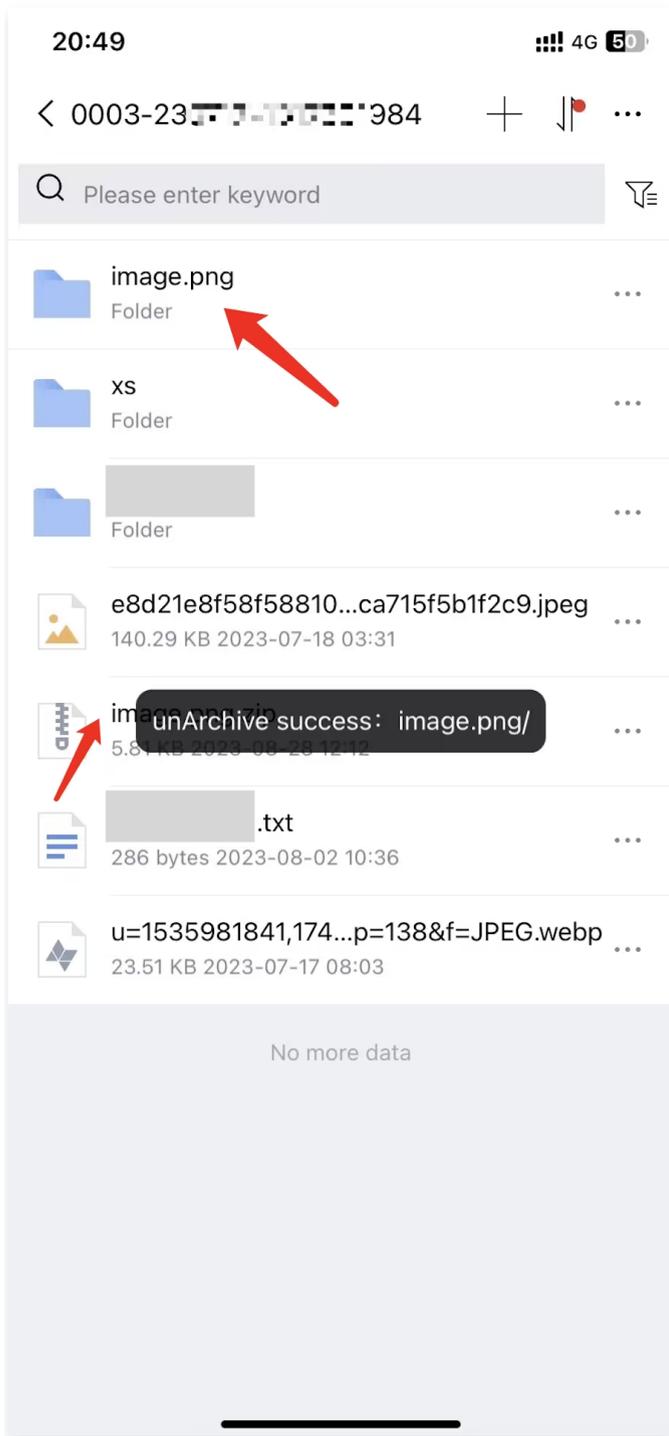
1. Navigate to the file list page and click the "..." next to the compressed file.



2. In the pop-up operation list, click **Online Unzip**.



3. Upon successful decompression, the extracted files will appear in the current directory.



## Sharing objects

COSBrowser provides the folder and file sharing feature for you to quickly collect data or share the data in a bucket with other users.

## Sharing folders

### Note

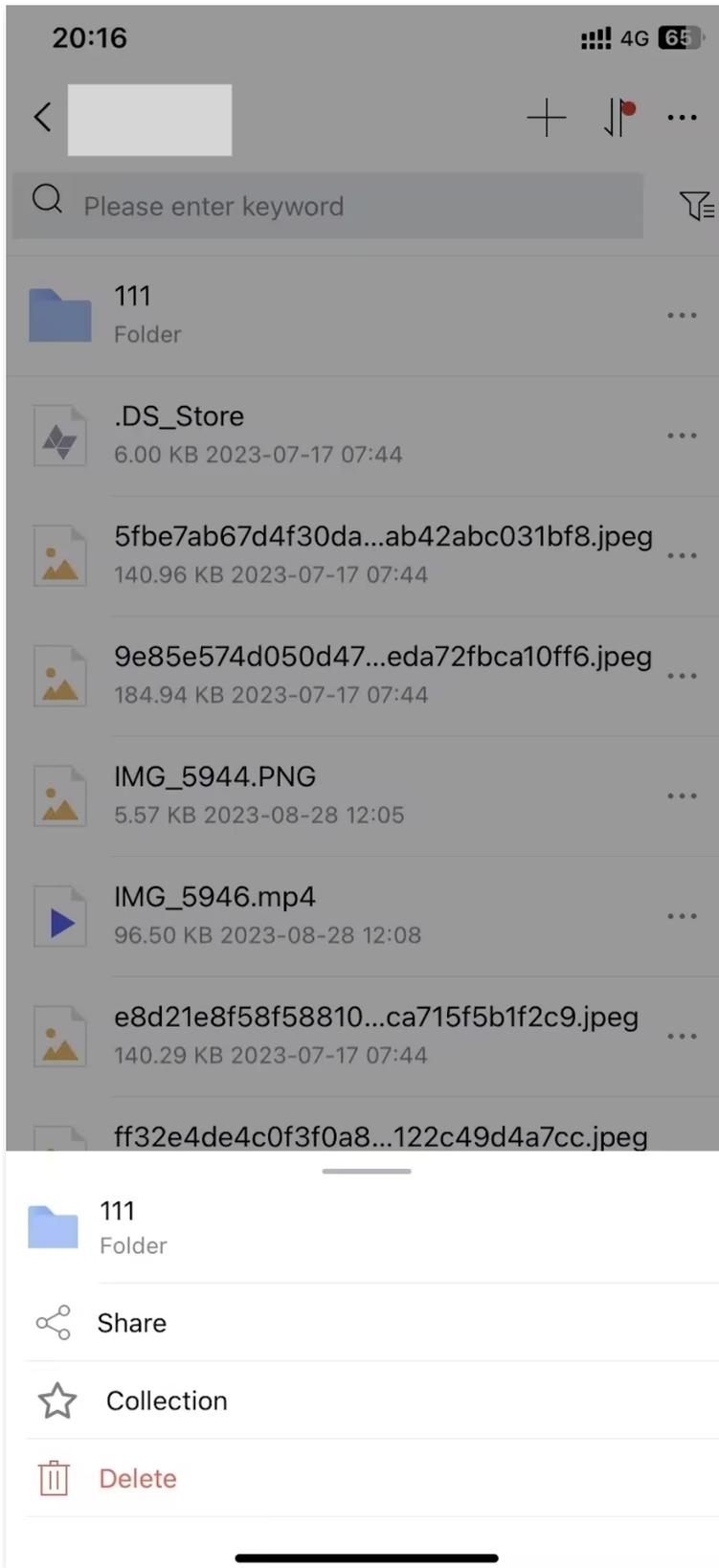
- You can only share a single folder but not multiple files.

- A sharing duration of up to 2 hours and up to 1.5 days can be set for the root account and a sub-account respectively.
- If multiple users share a folder, the file content may be hard to manage. In this case, we recommend you enable versioning for your bucket so that you can roll back to the desired historical version.

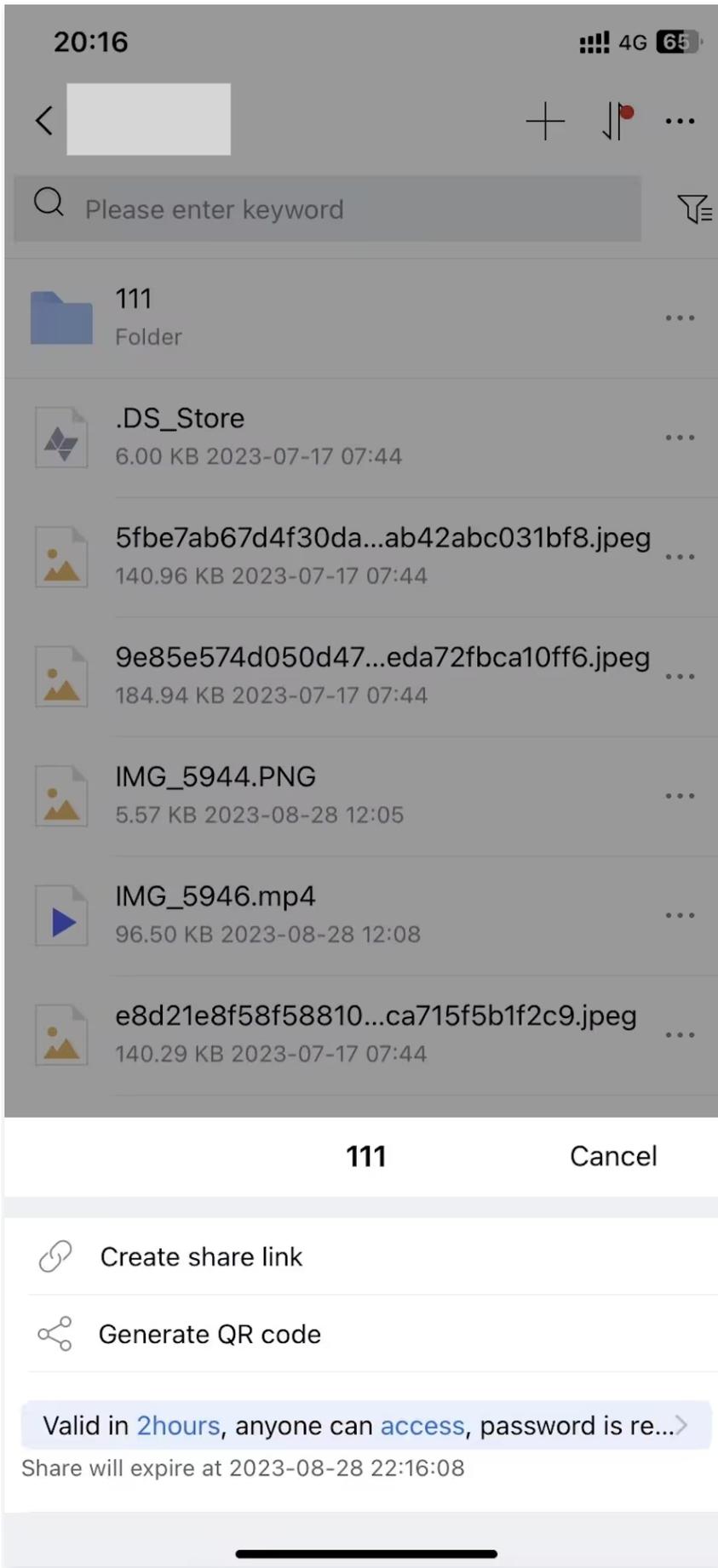
## Generating sharing QR code/link

The detailed steps are as follows:

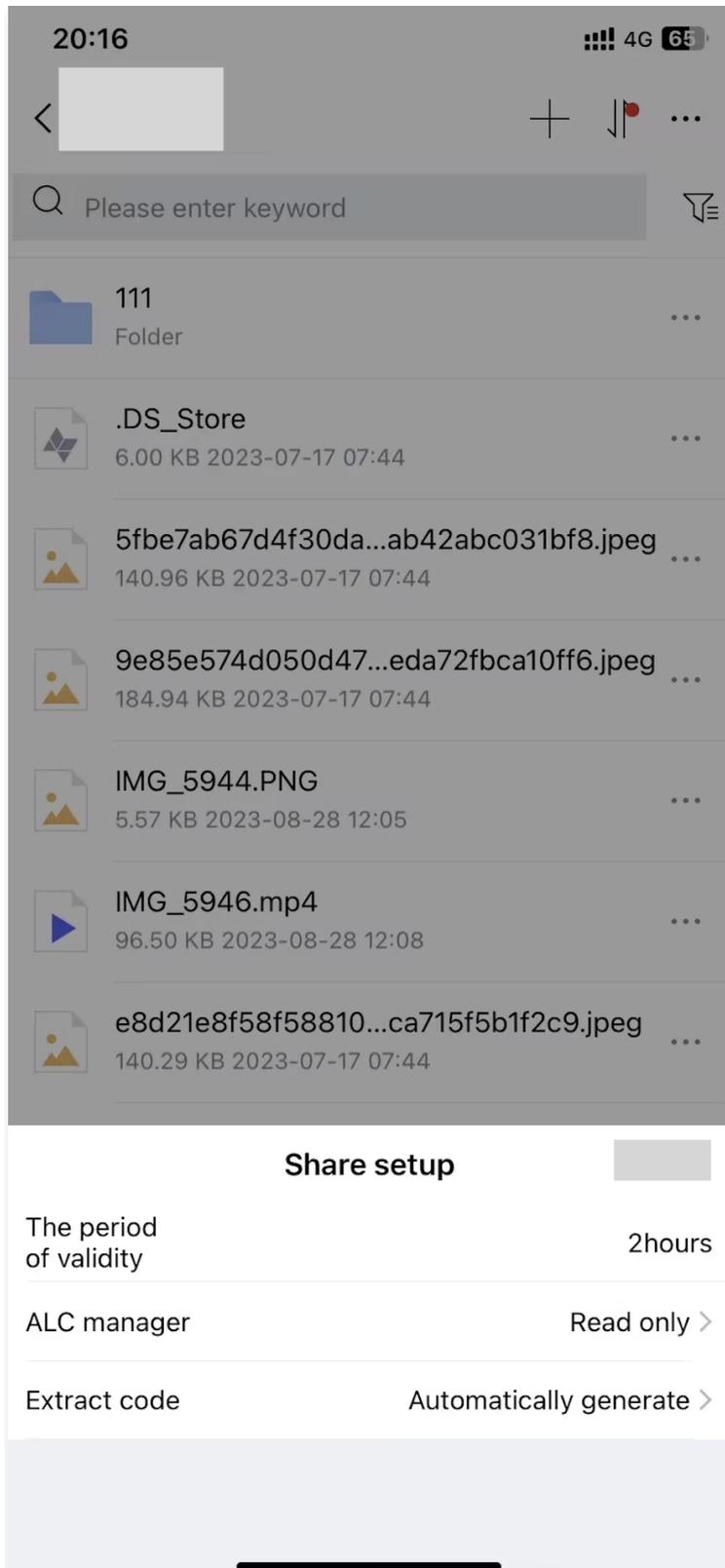
1. Click the "... " option to the right of the folder, a pop-up operation list will appear, then click Share.



2. In the pop-up sharing page, you can choose between QR code sharing or link sharing.

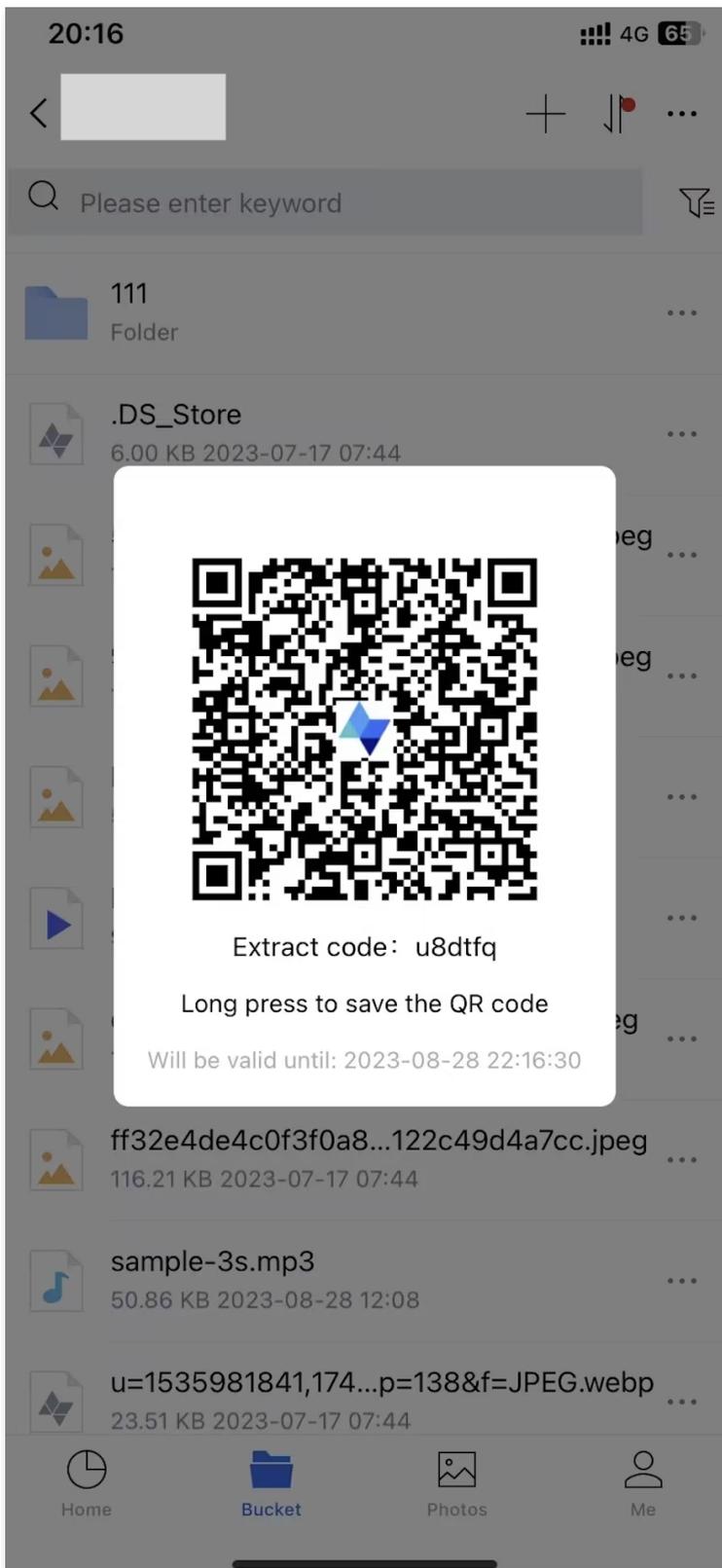


3. (Optional) Configure sharing parameters. The configuration options are as follows. You can leave them unset and keep the default settings.



- **Permission:** Sets the access permission for the shared folder.
  - Can view: Pulls the folder list and downloads files in the folder using the access URL.

- **Can edit:** Pulls the folder list and downloads files in the folder, upload files to the folder, and create folders using the access URL.
  - **Validity Period:** It is in minutes, hours, or days. The validity period for the root account and a sub-account is two hours and 24 hours respectively by default, which cannot be changed.
  - **Password:** A 6-character password automatically generated by the system. You can customize one as needed (numbers, letters, and symbols are supported).
4. Click **Generate Link** or **QR Code Sharing** to create the corresponding share link or QR code.



## Viewing folder shared by another user

You can access shared folders from others through mobile, desktop, and browser platforms.

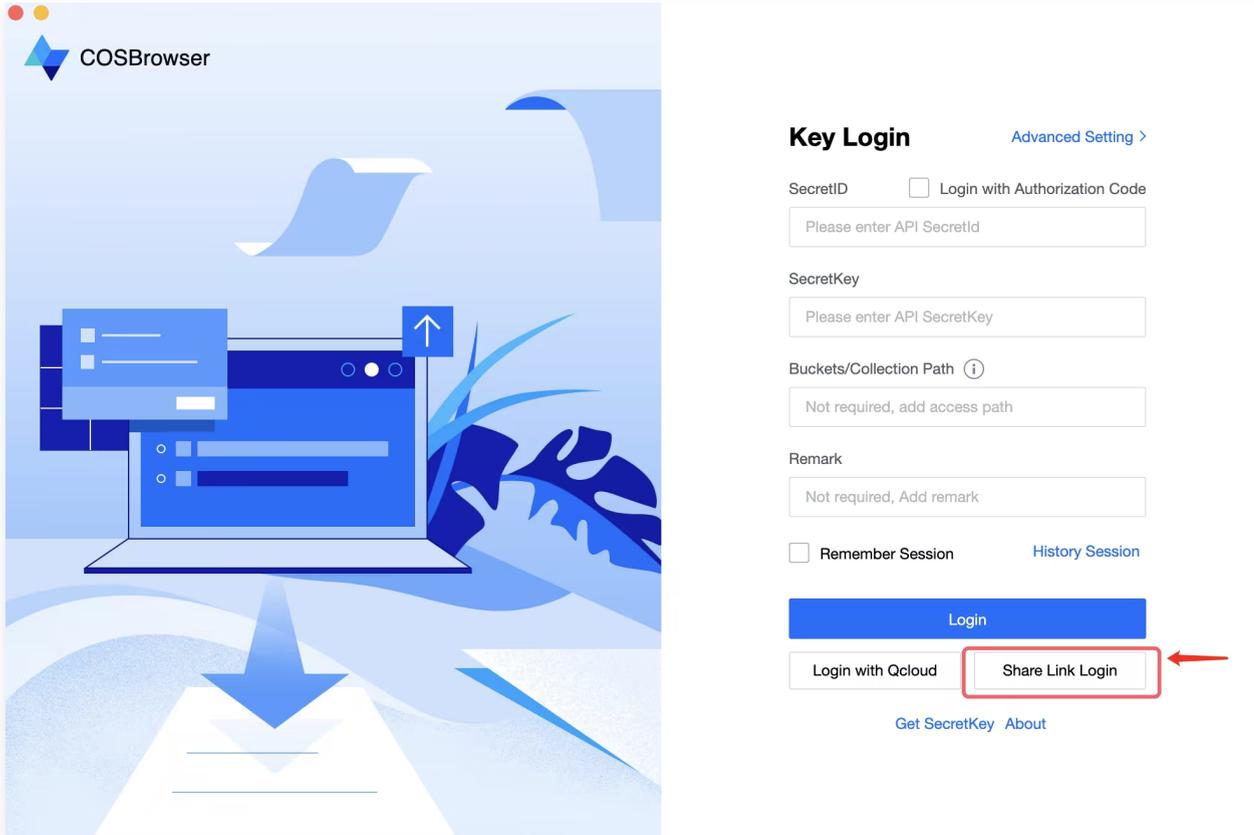
### Method 1: Viewing on Mobile

1. On the **Login page** or **Profile page**, click **Scan** to scan the shared QR code.

2. Enter the extraction code and click **Confirm** to access the shared folder.

## Method Two: Viewing on Desktop

1. On the login screen, click **Login with Shared Link**.



2. Enter the obtained URL address and extraction code, then log in.

### Share Link Login ✕

Share Link

Extraction code

Please enter the 6-digit extraction code

### Method 3. View in browser

1. Open a browser, enter the shared URL address, and open the link.
2. Enter the password and click **Extract** to enter the shared folder.

## Sharing files

Each file stored in COS can be accessed through a specific URL. You can generate a file URL with the specified domain name if you have set other domain names (such as CDN acceleration domain name and custom origin domain name) for the bucket. If a file is private-read, you can request a temporary signature to generate a temporary access URL with a certain validity period.

### Note

- If you log in with a temporary key, you cannot configure the validity period of the file link, which is 1 hour by default.
- If the file is public-read, the URL will not carry a signature and will be valid permanently. If the file is private-read, the URL will carry a temporary signature and will be valid for one hour.

## Instructions

1. Click on the file to enter the file details page.  
The configuration options are explained as follows:

- **Specify Domain:** Set the domain name for the URL (optional).
  - **Validity Period:** Set the validity period for the URL (optional).
2. After confirming that the configuration is correct, click **Generate Link**.
  3. Send the generated file URL.

## Renaming file

### Note

Folders cannot be renamed.

## Instructions

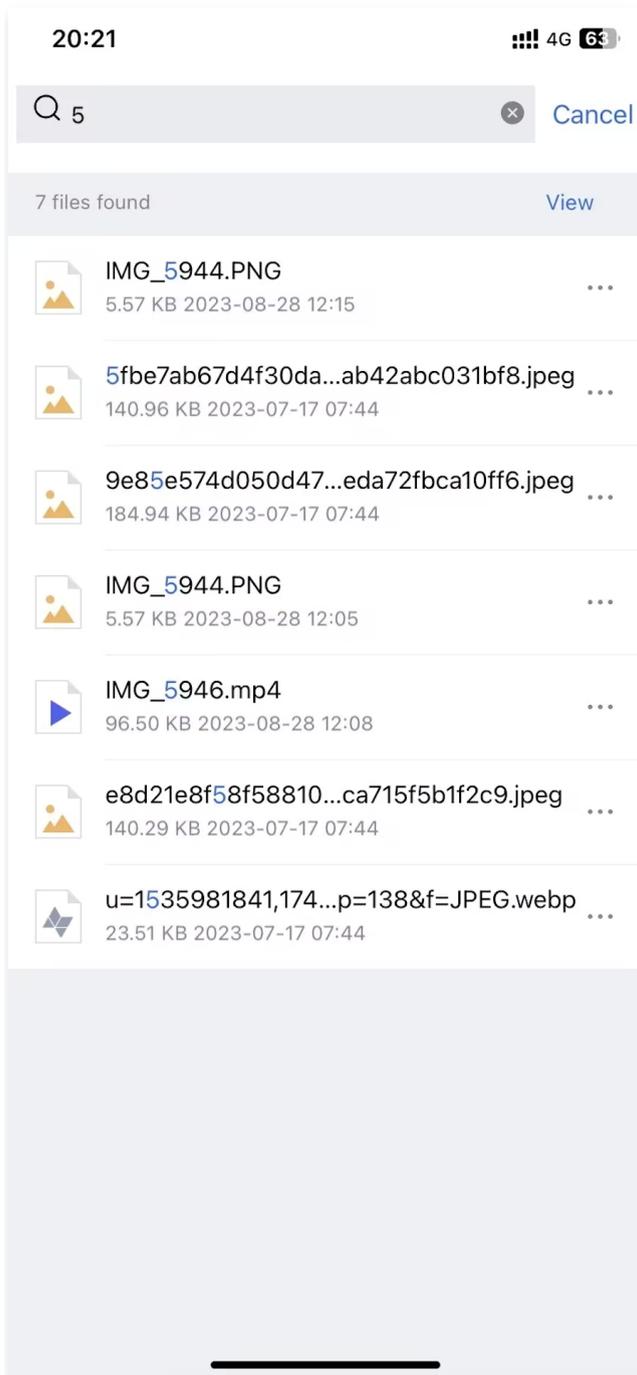
1. Click the "... " to the right of the file.
2. In the file operation list, click **Rename**.
3. In the pop-up window, enter the new folder name and click **OK**.  
If **Overwrite existing files** is checked, the original file will be overwritten. If unchecked, a new file will be created. You can also configure this globally in **Profile > Settings > Default Upload Options > Rename Duplicate Files**.

## Searching for objects

COSBrowser Mobile Version provides the fuzzy search and search by type features. You can use a prefix to search for files with that filename prefix in the current folder and its sub-folders. You can also specify the file type first to search for files in that type.

## Search by keyword

Enter the search keyword, and it will filter out files containing the keyword in the current folder and all subfolders.



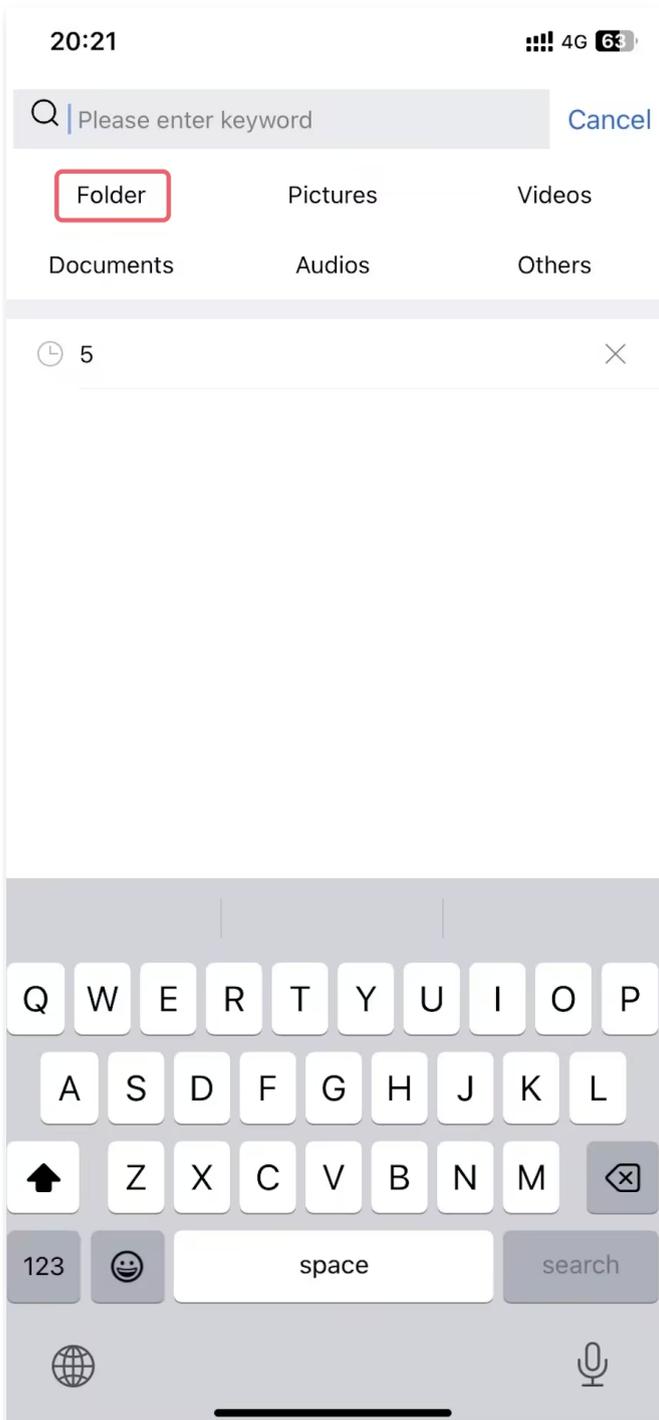
## Search by type

COSBrowser allows you to search for files by type, such as video, folder, audio, document, image, and others.

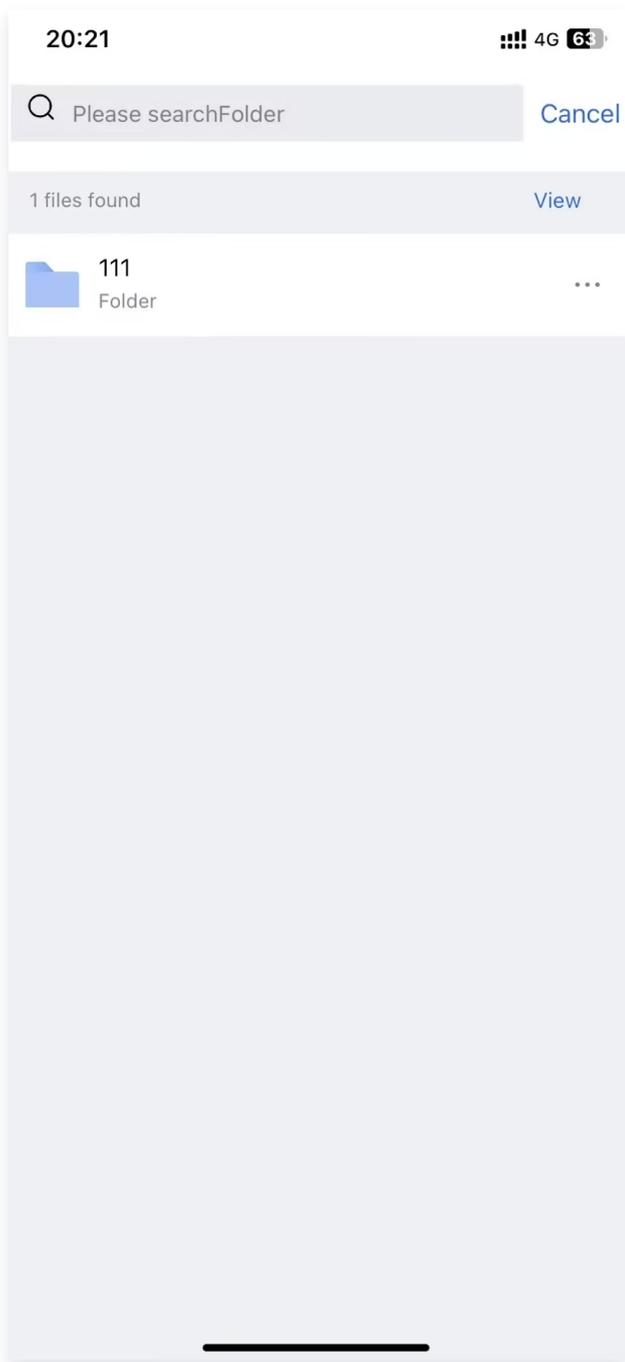
## Instructions

Taking folder search as an example:

1. After clicking the search box, click the folder.



2. All files of this type in the directory (folders in this example) will be listed.



## Sorting or Filtering Objects

COSBrowser allows you to sort and filter files in a bucket.

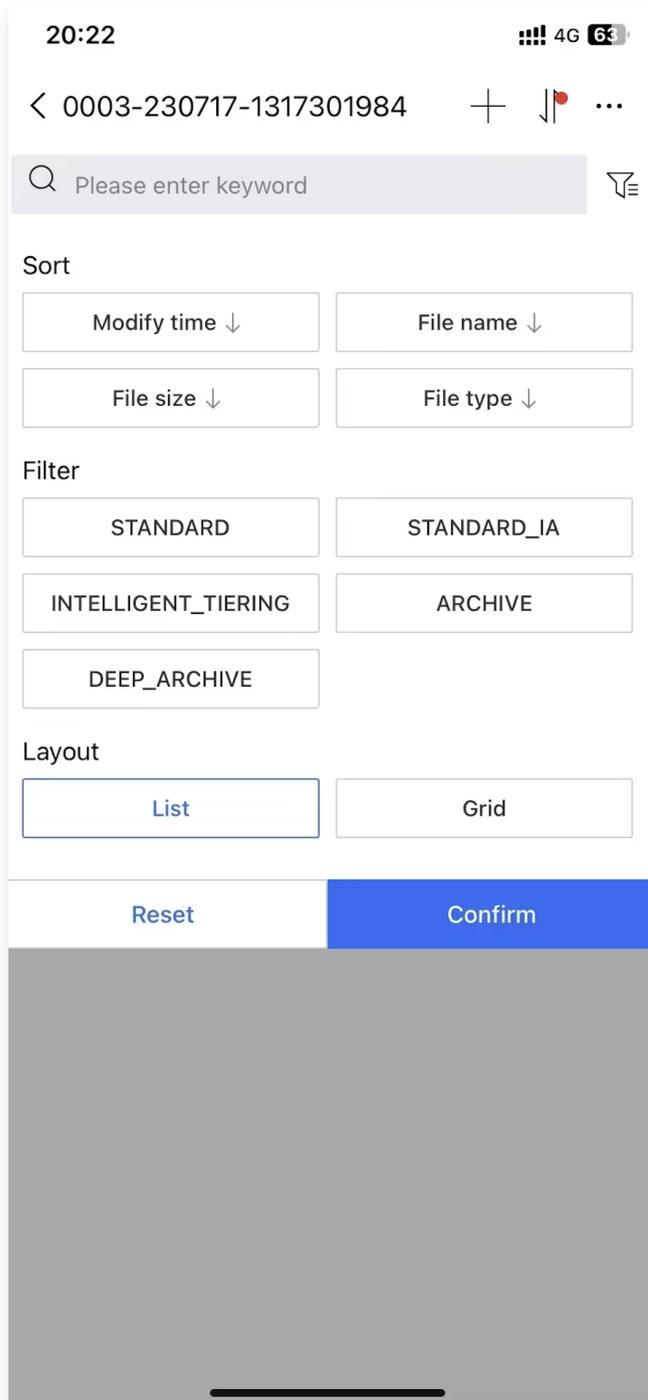
### Note

You can sort the file list by file name, size, or modification time, and filter by storage type.

## Instructions

1. Enter the file list page and click the filter and sort button on the right of the search box.

## 2. In the operation list, sort or filter files in the bucket.



## Managing File Permissions

COSBrowser allows you to set file access permissions, which takes priority over that for buckets.

### Note

- The object access permission is valid only when the access attempt is made via the default domain name. For any access attempt made via a CDN accelerated domain

name or custom domain name, the bucket access permission has the first priority.

- There are limits on the number of ACL rules. For more information, please see [Specifications and Limits](#).

## Modifying public permission

1. Click the target file to enter the **File Details** page.
2. Click **Permission Info** at the top to enter the permission list page.
3. Click **Public Permission** to modify the access permissions of the file.

## Setting user permissions

1. Click the target file to enter the **File Details** page.
2. Click **Permission Info** at the top to enter the permission list page.
3. Click on **Public Permissions** to modify the user permissions for the file.  
You can add a new user permission by clicking **Add User**. After adding a new user permission, you can swipe left to edit or delete a user permission.

# Resource Packs

Last updated: 2023-09-07 12:18:27

## Feature Overview

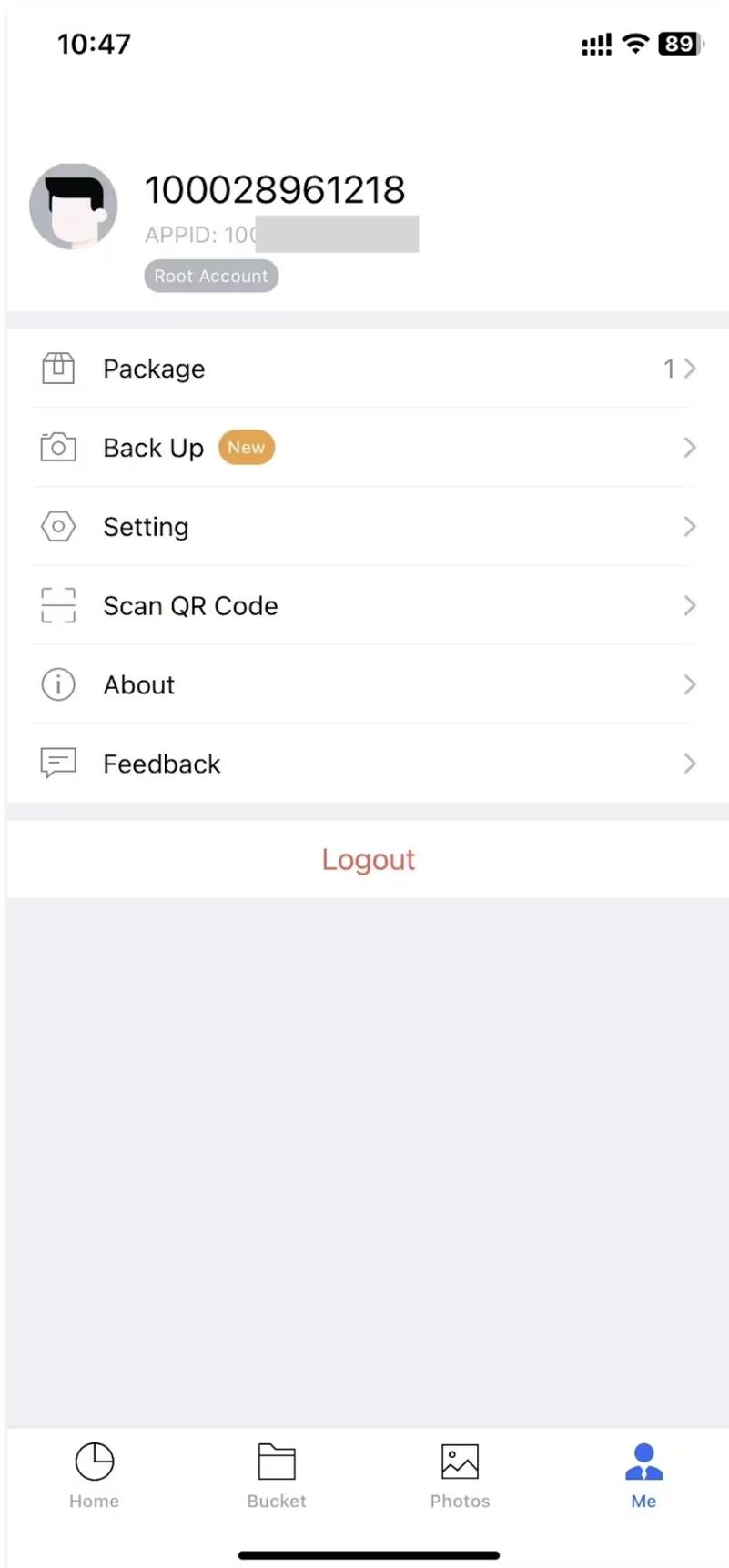
COSBrowser Mobile Version allows you to view the list of resource packs, purchase, renew, and upgrade resource packs, and perform other operations.

### Note

- For more information on the types, purchase, validity period, and effective scope of resource packs, see [Resource Pack Overview](#).
- To purchase, renew, or upgrade a resource pack on COSBrowser Mobile Version, you need to be redirected to the Tencent Cloud Assistant mini program. Before performing the operation, make sure that you are logged in to COSBrowser Mobile Version and the Tencent Cloud Assistant mini program with the same Tencent Cloud account.

## Viewing the list of resource packs

Navigate to **Profile > Resource Packs** to view the existing resource packs under your current account.



## Purchasing a resource pack

Follow the steps below:

1. Navigate to **Profile > Resource Packs**, click on **Purchase Resource Pack**, and you will be redirected to the resource pack purchase page in the mini program.
2. Select the appropriate resource pack according to your business needs, click **Buy Now**, and follow the prompts to complete the payment.

## Renewing a resource pack

If a resource pack will expire soon, you can manually or automatically renew it to extend its validity period.

Follow the steps below:

1. Navigate to **Profile > Resource Packs** and click on the "..." icon next to the resource pack that needs renewal.
2. In the pop-up operation list, click **Renew**, and you will be redirected to the resource pack renewal page of the Tencent Cloud Assistant mini program.
3. Select the renewal duration, click **Buy Now**, and make the payment as prompted.

### Note

If there is no option to renew, it means that the resource pack does not support renewal, and you need to make a new purchase.

## Upgrading a resource pack

COSBrowser Mobile Version allows you to upgrade a resource pack. If the specification of an existing resource pack no longer meets your actual business needs, click **Upgrade** to upgrade the specification.

Follow the steps below:

1. Go to **Profile > Resource Packs** and click the ellipsis (...) on the right side of the resource pack you want to upgrade.
2. In the pop-up operation list, click **Upgrade**, and you will be redirected to the resource pack upgrade page of the Tencent Cloud Assistant mini program.
3. Select the new specification, click **Buy Now**, and make the payment as prompted.

### Note

If there is no upgrade option available, it means that the resource pack does not support upgrading, and you need to make a new purchase.

# COSCMD

Last updated: 2025-07-17 15:26:28

## Note

COSCMD enables you to perform batch object operations such as upload/download/delete by using simple command lines.

## Operating Environment

### System environment

Windows, Linux, and macOS.

#### Note:

- Local characters should use UTF-8 encoding. Otherwise, exceptions will occur when you operate on Chinese files.
- Ensure that the local time is in sync with UTC. If there is a large deviation between the two, COSCMD might not function properly.

## Software requirements

- Python 2.7 and Python 3
- Latest version of pip

#### Note:

It is recommended that users install the Python version with integrated pip directly.

## Installation and configuration

- For more information about the installation and configuration of the environment, see [Python](#).
- For more information about the installation and configuration of pip, see [Installation](#).

## Download and Installation

You can install COSCMD in the following three ways:

### Method 1:

#### 1. Installing with pip

Run the `pip` command to install COSCMD:

```
pip install coscmd
```

After the installation is complete, you can run the `-v` or `--version` command to view the version information.

**Note:**

After installing on Windows, add both `C:\python_install_dir;` and `C:\python_install_dir\Scripts` to the environment variables.

## 2. Upgrading with pip

After the installation is complete, you can run the following command to upgrade COSCMD:

```
pip install coscmd -U
```

## Method 2: Source Code Installation (Not Recommended)

You can click [here](#) to download the source code.

```
git clone https://github.com/tencentyun/coscmd.git
cd coscmd
python setup.py install
```

**Note:**

When using Python 2.6, pip installation of dependency libraries may fail. It is recommended to use an alternative method for installation.

## Method 3: Offline Installation

**Note:**

Ensure that both machines have the same Python version to avoid installation failures.

```
# Run the following command on a machine with internet access:
mkdir coscmd-packages
pip download coscmd -d coscmd-packages
tar -czvf coscmd-packages.tar.gz coscmd-packages
```

```
# After copying the installation package to a machine without internet
access, run the following command:
tar -xzvf coscmd-packages.tar.gz
pip install coscmd --no-index -f coscmd-packages
```

## Parameter Configuration

### Viewing the help option

Run the `-h` or `--help` command to view the information and usage of COSCMD.

```
coscmd -h
```

The help information is as follows:

```
usage: coscmd [-h] [-d] [-s] [-b BUCKET] [-r REGION] [-c CONFIG_PATH]
             [-l LOG_PATH] [--log_size LOG_SIZE]
             [--log_backup_count LOG_BACKUP_COUNT] [-v]

{config,upload,download,delete,abort,copy,move,list,listparts,info,resto
re,signurl,createbucket,deletebucket,putobjectacl,getobjectacl,putbucket
acl,getbucketacl,putbucketversioning,getbucketversioning,probe}
...

an easy-to-use but powerful command-line tool. try 'coscmd -h' to get
more
informations. try 'coscmd sub-command -h' to learn all command usage,
likes
'coscmd upload -h'

positional arguments:

{config,upload,download,delete,abort,copy,move,list,listparts,info,resto
re,signurl,createbucket,deletebucket,putobjectacl,getobjectacl,putbucket
acl,getbucketacl,putbucketversioning,getbucketversioning,probe}
    config                Config your information at first
    upload                Upload file or directory to COS
    download              Download file from COS to local
    delete               Delete file or files on COS
```

```
abort          Aborts upload parts on COS
copy           Copy file from COS to COS
move           move file from COS to COS
list           List files on COS
listparts      List upload parts
info           Get the information of file on COS
restore        Restore
signurl        Get download url
createbucket   Create bucket
deletebucket   Delete bucket
putobjectacl   Set object acl
getobjectacl   Get object acl
putbucketacl   Set bucket acl
getbucketacl   Get bucket acl
putbucketversioning
               Set the versioning state
getbucketversioning
               Get the versioning state
probe          Connection test
```

optional arguments:

```
-h, --help      show this help message and exit
-d, --debug     Debug mode
-s, --silence   Silence mode
-b BUCKET, --bucket BUCKET
                Specify bucket
-r REGION, --region REGION
                Specify region
-c CONFIG_PATH, --config_path CONFIG_PATH
                Specify config_path
-l LOG_PATH, --log_path LOG_PATH
                Specify log_path
--log_size LOG_SIZE specify max log size in MB (default 1MB)
--log_backup_count LOG_BACKUP_COUNT
                specify log backup num
-v, --version   show program's version number and exit
```

In addition, you can enter `-h` after each command (with no parameter appended) to see how to use the command. For example:

```
coscmd upload -h // View how to use the upload command
```

## Generating a configuration file via the `config` command

### Note:

- We recommend using the tool with [temporary keys](#) to enhance security through temporary authorization. When applying for temporary keys, please follow the [principle of least privilege](#) to prevent leakage of resources beyond the target bucket or object.
- If you must use a permanent key, we recommend you follow the [Notes on Principle of Least Privilege](#) to limit the scope of permission on the permanent key.

The `config` command can automatically generate a configuration file at `~/.cos.conf`. The specific command format is as follows:

```
coscmd config [OPTION]...<FILE>...
    [-h] --help
    [-a] <SECRET_ID>
    [-s] <SECRET_KEY>
    [-t] <TOKEN>
    [-b] <BucketName-APPID>
    [-r] <REGION> | [-e] <ENDPOINT>
    [-m] <MAX_THREAD>
    [-p] <PART_SIZE>
    [--do-not-use-ssl]
    [--anonymous]
```

### Note:

Fields within "[]" are optional, while those within "<>" require input.

The following table describes these parameters:

Option	Description	Valid Value	Required
-a	Key ID, which can be obtained on the <a href="#">API Key Management Page</a> on the console	String	Required
-s	Key, which can be obtained on the <a href="#">API Key Management Page</a> on the console	String	Required
-t	Temporary key token, which needs to be specified in the <code>x-cos-security-token</code> header when a temporary key is used.	String	Not required
-b	Specify the bucket name in the format <code>BucketName-APPID</code> , as described in <a href="#">Naming Conventions</a> . For the initial configuration, create a bucket in the COS console to be used for tool configuration.	String	Required
-r	Region of the bucket. For more information, see <a href="#">Regions and Access Endpoints</a> .	String	Required
-e	Set the request ENDPOINT. After setting the ENDPOINT parameter, the REGION parameter will be invalid. If you are using the default domain name, the configuration format is <code>cos.&lt;region&gt;.myqcloud.com</code> ; if you are using the global acceleration domain name, the configuration is <code>cos.accelerate.myqcloud.com</code> .	String	Not required
-m	Maximum number of threads for multithreaded operation (default is 5). If the file upload is slow, you can increase this value. The number of threads depends on the machine's performance. Assuming the machine supports a maximum of 30 threads, you can adjust the concurrent threads to 30 to fully utilize the machine performance. The command is <code>coscmd config -m 30</code> .	Number	Not required
-p	The block size for chunked operations (in MB, default is 1 MB, range is 1 - 1000). If the file upload is slow, you can increase this value accordingly.	Number	Not required
--do-			Not

<code>not-use-ssl</code>	Uses the HTTP protocol instead of HTTPS	String	required
<code>--anonymous</code>	Anonymous operation (i.e. one that does not carry a signature)	String	Not required

The following sample shows how to use the `config` command:

```
coscmd config -a ***** -s
***** -b configure-bucket-1250000000
-r ap-chengdu
```

## Generating a configuration file

COSCMD will first read the necessary information from the configuration file before running. By default, COSCMD will read configuration items from `~/.cos.conf`.

### Note:

Before configuring, create a bucket (e.g., `configure-bucket-1250000000`) in the COS console and generate the necessary key information.

The following is a sample configuration file:

```
[common]
secret_id = *****
secret_key = *****
bucket = configure-bucket-1250000000
region = ap-chengdu
max_thread = 5
part_size = 1
retry = 5
timeout = 60
schema = https
verify = md5
anonymous = False
```

### Note:

- The `timeout` item in the configuration file, in seconds, indicates the timeout period for reading and writing data.

- In the configuration file, the `schema` option can be set to either "http" or "https", with "https" being the default value.
- In the configuration file, the `anonymous` option can be set to True or False, indicating whether to use anonymous mode, i.e., keeping the signature empty.
- For more information about the parameter description, run the `coscmd config -h` command.

## Common Commands

### Using commands to specify a bucket and region

If you do not specify the bucket and region in the commands, the commands take effect for the bucket that is used to configure COSCMD. To perform operations on another bucket, you need to specify the bucket and the region where the bucket resides.

#### Note:

- Specify the bucket name using the `-b <BucketName-APPID>` parameter, adhering to the BucketName-APPID naming format.
- Specify the region using `-r <region>` to set the bucket's location.

#### • Command format

```
coscmd -b <BucketName-APPID> -r <region> <action> ...
```

#### • Sample: creating a bucket named `examplebucket` that resides in the Beijing region

```
coscmd -b examplebucket-1250000000 -r ap-beijing createbucket
```

#### • Sample: uploading "picture.jpg" from D drive to the `examplebucket` bucket

```
coscmd -b examplebucket-1250000000 -r ap-beijing upload D:/picture.jpg  
/
```

## Specifying the configuration file and log file paths

If you do not specify the configuration file's path, `~/.cos.conf` will be used by default. Similarly, if you do not specify the log file's path, `~/.cos.log` will be used by default.

#### Note:

- Use the `-c <conf_path>` parameter to specify the configuration file path, and COSCMD will read the configuration information from this path during runtime.
- Specify the log path using the `-l <log_conf>` parameter, and COSCMD will output logs generated during the process to the log file in this path.

- **Command format**

```
coscmd -c <conf_path> -l <log_conf> <action> ...
```

- **Sample: setting the configuration file's path to `/data/home/cos_conf` and the log path to `/data/home/cos_log`, and creating a bucket named `examplebucket` in the Beijing region.**

```
coscmd -c /data/home/cos_conf -l /data/home/cos_log -b examplebucket-1250000000 -r ap-beijing createbucket
```

## Running commands in debugging mode

If `-d` or `-debug` is added before a command, detailed operation information will be displayed when executing the command, as shown in the example below:

- **Command format**

```
coscmd -d upload <localpath> <cospath>
```

- **Sample: outputting detailed information upon the upload**

```
coscmd -d upload -rs D:/folder/ /
```

## Running commands in silence mode

You can prefix `-s` or `--silence` in each command so that no message will be output.

**Note:**

To run this command, the version should be at least 1.8.6.24.

- **Command format**

```
coscmd -s upload <localpath> <cospath>
```

- Sample

```
coscmd -s upload D:/picture.jpg /
```

## Common Bucket Commands

### Create a bucket.

**Note:**

When executing the create bucket command, include the parameter `-b <BucketName-APPID>` to specify the bucket name and `-r <Region>` to specify the region. If you directly execute `coscmd createbucket` without specifying the bucket name and region, an error will occur because it is equivalent to creating an existing bucket (i.e., the bucket specified during configuration).

- Command format

```
coscmd -b <BucketName-APPID> createbucket
```

- Sample: creating a bucket named `examplebucket` that resides in the Beijing region

```
coscmd -b examplebucket-1250000000 -r ap-beijing createbucket
```

### Deleting a bucket

**Note:**

`coscmd deletebucket` usage is only valid for the bucket specified during configuration. It is recommended to use it with `-b <BucketName-APPID>` to specify the bucket and `-r <region>` to specify the region.

- Command format

```
coscmd -b <BucketName-APPID> deletebucket
```

- Sample: deleting empty buckets

```
coscmd -b examplebucket-1250000000 -r ap-beijing deletebucket
```

- Sample: forcibly deleting a non-empty bucket

```
coscmd -b examplebucket-1250000000 -r ap-beijing deletebucket -f
```

**Note:**

Using the `-f` parameter will forcefully delete the bucket, including all files, historical folders after enabling version control, and fragments generated by uploads. Please proceed with caution.

## Common Object Commands

### Uploading a file

- Command syntax for uploading a file

```
coscmd upload <localpath> <cospath>
```

**Note:**

Please replace the parameters within "<>" with the local file path (localpath) you want to upload and the storage path on COS (cospath).

- Sample: uploading "picture.jpg" in D drive to the "doc" folder of COS

```
coscmd upload D:/picture.jpg doc/
```

- Sample: uploading "picture.jpg" in the "doc" folder in D drive to the "doc" folder of COS

```
coscmd upload D:/doc/picture.jpg doc/
```

- Sample: uploading a file to the ARCHIVE storage class to the "doc" directory of COS

```
coscmd upload D:/picture.jpg doc/ -H '{"x-cos-storage-class': 'Archive'}"
```

**Note:**

When using the `-H` parameter to set the HTTP header, ensure the format is JSON, for example: `coscmd upload -H '{"x-cos-storage-class': 'Archive', 'Content-Lan`

uage': 'zh-CN'}" <localpath> <cospath> . For more headers, refer to the [PUT Object](#) documentation.

- Sample: setting meta attributes and uploading a file to the “doc” folder of COS

```
coscmd upload D:/picture.jpg doc/ -H '{"x-cos-meta-example': 'example'}"
```

- Sample: Set the upload speed limit to 800Kb/s and upload the file file.zip from the doc folder on Drive D to the doc directory on COS.

```
coscmd upload D:/doc/file.zip doc/ -H '{"x-cos-traffic-limit': '819200'}"
```

**Note:**

Specify the request header `x-cos-traffic-limit` via the `-H` parameter to limit the speed. The speed range is 819200 to 838860800 (in bit/s), that is, 800 Kb/s to 800 Mb/s. If a value is not within this range, a 400 error will be returned.

## Uploading a folder

- Command syntax for uploading a folder

```
coscmd upload -r <localpath> <cospath>
```

**Note:**

For Windows users, it is recommended to use the COSCMD upload command in the built-in cmd tool or PowerShell. Other tools (such as git bash) may have different command path parsing strategies than PowerShell, which could result in files being uploaded to incorrect paths.

- Sample: uploading the "doc" folder in D drive to the root directory of COS

```
coscmd upload -r D:/doc /
```

- Sample: uploading the "doc" folder in D drive to the "doc" folder of COS

```
coscmd upload -r D:/doc doc
```

- Sample: uploading files synchronously (files with the same name, MD5 checksum, and file size will be skipped)

```
coscmd upload -rs D:/doc doc
```

**Note:**

Using the `-s` parameter enables synchronous uploads, skipping files with matching MD5 values (the original files in COS must be uploaded using COSCMD version 1.8.3.2 or later, which includes the `x-cos-meta-md5` header by default).

- Sample: uploading files synchronously (files with the same name and file size will be skipped)

```
coscmd upload -rs --skipmd5 D:/doc doc
```

**Note:**

Using the `-s` parameter enables synchronous uploads, and when combined with the `--skipmd5` parameter, it compares the file sizes of files with the same name. If the sizes are identical, the upload is skipped.

- Sample: uploading the folder synchronously and deleting files that are deleted in the "doc" folder in D drive

```
coscmd upload -rs --delete D:/doc /
```

**Note:**

For example, if the files in the Drive D doc folder have been synchronously uploaded to the doc path in COS on the same day, and the source and destination are consistent. On the next day, if the user deletes file A from the Drive D doc folder, executing the above command will delete file A from the doc path in COS, ultimately keeping the files in the Drive D doc folder consistent with those in the doc path in COS.

- Sample: ignoring .txt and .doc files in the "doc" folder in D drive

```
coscmd upload -rs D:/doc / --ignore *.txt,*.doc
```

- Sample: ignoring .txt files in the "doc" folder in D drive

```
coscmd upload -rs D:/doc / --ignore "*.txt"
```

#### Note:

- When uploading folders, you can ignore certain types of files by using the `--ignore` parameter, or filter certain types of files by using `--include`. Multiple shell wildcard rules (separated by commas `,`) are supported. When ignoring a certain suffix, it must be enclosed in `" "`.
- If you want to use `--ignore` to filter all files in a particular folder, you need to use an absolute path and use `" "` to enclose the path, for example, `coscmd upload -rs D:/doc / --ignore "D:/doc/ignore_folder/*"`.

- Sample: uploading .txt and .doc files in the "doc" folder in D drive

```
coscmd upload -rs D:/doc / --include *.txt,*.doc
```

- Sample: uploading .txt files in the "doc" folder in D drive

```
coscmd upload -rs D:/doc / --include "*.txt"
```

#### Note:

- For files larger than 10MB, COSCMD uses multipart upload with the same command as simple upload: `coscmd upload <localpath> <cospath> .`
- COSCMD supports checkpoint restart to resume the upload of large files. When the multipart upload of a large file fails, only the failed parts will be uploaded when the operation is resumed instead of starting over from scratch (please ensure that the directory and content of the re-uploaded file are consistent with the uploaded directory).
- COSCMD performs MD5 verification on each part during multipart upload.
- By default, COSCMD uploads include the `x-cos-meta-md5` header, with the value being the file's MD5 hash. If the `--skipmd5` parameter is used, this header will not be included.

## Querying a file list

The query command is as follows:

- Command format.

```
coscmd list <cospath>
```

- Sample: recursively querying the list of all files prefixed with "doc/" in this bucket.

```
coscmd list doc/
```

- Sample: recursively querying the file list, number of files, and the file sizes of a bucket.

```
coscmd list -ar
```

- Sample: recursively querying the list of all files prefixed with "examplefolder".

```
coscmd list examplefolder/ -ar
```

- Sample: querying the historical versions of all files in a bucket.

```
coscmd list -v
```

### Note:

- Replace the parameters in "<>" with the COS path (cospath) of the file list you want to query. If `<cospath>` is empty, the default query is for the root directory of the current storage bucket.
- Use `-a` to query all files.
- Use `-r` to query files recursively. The number and total size of the files are listed at the end of the returned result.
- Use `-n num` to set the maximum number of files to query.

## Viewing the file information

The command is as follows:

- Command format.

```
coscmd info <cospath>
```

- Sample: viewing the metadata of "doc/picture.jpg".

```
coscmd info doc/picture.jpg
```

**Note:**

Please replace the parameters in "<>" with the COS file path (cospath) you want to display.

## Downloading a file or folder

### Command syntax for downloading a file

```
coscmd download <cospath> <localpath>
```

**Note:**

Replace the parameters in "<>" with the COS file path (cospath) you want to download and the local storage path (localpath).

- Sample: downloading the "doc/picture.jpg" file in COS to "D:/picture.jpg" .

```
coscmd download doc/picture.jpg D:/picture.jpg
```

- Sample: downloading the "doc/picture.jpg" file in COS to drive D.

```
coscmd download doc/picture.jpg D:/
```

- Sample: downloading a specified version of "picture.jpg" to drive D.

```
coscmd download picture.jpg --versionId MTg0NDUxMzc2OTM4NTEExNTg7Tjg  
D:/
```

### Command syntax for downloading a folder

```
coscmd download -r <cospath> <localpath>
```

- Sample: downloading the "doc" folder to "D:/folder/doc".

```
coscmd download -r doc D:/folder/
```

- Sample: downloading files in the root directory while ignoring those in the `doc` directory that is under the root directory.

```
coscmd download -r / D:/ --ignore "doc/*"
```

- Sample: downloading all files in the root directory of the current bucket and overwriting local files.

```
coscmd download -rf / D:/examplefolder/
```

**Note:**

If a file with the same name exists locally, the download will fail. Use the `-f` parameter to overwrite the local file.

- Sample: synchronously downloading all files in the root directory of the current bucket while skipping those with the same filename and MD5 checksum.

```
coscmd download -rs / D:/examplefolder
```

**Note:**

Use the `-s` or `--sync` parameter to skip downloading files that already exist locally with the same content (provided that the files were uploaded using the COSCMD upload interface and have the `x-cos-meta-md5` header).

- Sample: synchronously downloading all files in the root directory of the current bucket while skipping those with the same filename and file size.

```
coscmd download -rs --skipmd5 / D:/examplefolder
```

- **Sample: Synchronous download and delete "files deleted from the doc path in the COS bucket".**

```
coscmd download -rs --delete / D:/doc /
```

**Note:**

For example, if the files in the D drive doc folder and the files under the doc path in COS have been synchronously uploaded on the same day, making the source end and target end identical. The next day, if the user deletes file A under the doc path in COS, executing the above command will delete file A from the D drive doc folder, ultimately keeping the files in the D drive doc folder and the files under the doc path in COS consistent.

- **Sample: ignoring .txt or .doc files.**

```
coscmd download -rs / D:/examplefolder --ignore *.txt,*.doc
```

- **Sample: ignoring .txt files.**

```
coscmd download -rs / D:/examplefolder --ignore "*.txt"
```

**Note:**

- When uploading folders, you can ignore certain types of files by using the `--ignore` parameter, or filter certain types of files by using `--include`. Multiple shell wildcard rules (separated by commas `,`) are supported. When ignoring a certain suffix, it must be enclosed in `" "`.
- To use `--ignore` to filter all files within a specific directory, you need to use an absolute path and enclose it in `" "`. For example, `coscmd upload -rs D:/doc / --ignore "D:/doc/ignore_folder/*"`.

- **Sample: filtering .txt and .doc files.**

```
coscmd download -rs / D:/examplefolder --include *.txt,*.doc
```

- **Sample: filtering .txt files.**

```
coscmd download -rs / D:/examplefolder --include "*.txt"
```

**Note:**

The old mget interface has been deprecated; the download interface uses chunked downloading. Please use the download interface.

## Getting signed download URLs

- Command format.

```
coscmd signurl <cospath>
```

- Sample: generating a signed URL for "doc/picture.jpg".

```
coscmd signurl doc/picture.jpg
```

- Sample: generating a signed URL that is effective for 100 seconds for "doc/picture.jpg".

```
coscmd signurl doc/picture.jpg -t 100
```

**Note:**

- Replace the parameters in "<>" with the COS path (cospath) of the file for which you want to obtain the download URL.
- Use `-t time` to set the valid duration of the signature in the URL (in seconds), with a default of 10000s.

## Deleting a file or folder

### Command syntax for deleting a file

```
coscmd delete <cospath>
```

**Note:**

Please replace the parameters in "<>" with the COS file path (cospath) you want to delete. The tool will prompt the user to confirm the deletion operation.

- Sample: deleting the “doc/exampleobject.txt” file.

```
coscmd delete doc/exampleobject.txt
```

- Sample: deleting files with version IDs.

```
coscmd delete doc/exampleobject.txt --versionId  
MTg0NDUxMzc4ODAzNTgyMTErEWN
```

## Command syntax for deleting a folder

```
coscmd delete -r <cospath>
```

- Sample: deleting the `doc` folder.

```
coscmd delete -r doc
```

- Sample: deleting the `folder/doc` folder.

```
coscmd delete -r folder/doc
```

- Sample: deleting all files with version IDs in the `doc` directory.

```
coscmd delete -r doc/ --versions
```

### Note:

- You need to enter `y` to confirm a batch delete operation. You can skip this step if the `-f` parameter is used.
- Note that the delete folder command will delete the current folder as well as the files in it. To delete a versioning-enabled file, you need to specify a version ID.

## Viewing incomplete multipart uploads

- Command format.

```
coscmd listparts <cospath>
```

- Sample: listing incomplete multipart uploads prefixed with "doc/".

```
coscmd listparts doc/
```

## Aborting incomplete multipart uploads

- Command format.

```
coscmd abort
```

- Sample: aborting all incomplete multipart uploads.

```
coscmd abort
```

## Copying a file or folder

### Command syntax for copying a file

```
coscmd copy <sourcepath> <cospath>
```

- Sample (intra-bucket replication): copying "picture.jpg" in the `examplebucket-1250000000` bucket to the "doc" folder.

```
coscmd -b examplebucket-1250000000 -r ap-chengdu copy examplebucket-1250000000.ap-chengdu.myqcloud.com/picture.jpg doc/
```

- Sample (cross-bucket replication): copying "doc/picture.jpg" in the `examplebucket2-1250000000` bucket to "doc/examplefolder/" in the `examplebucket1-1250000000` bucket.

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000000.ap-beijing.myqcloud.com/doc/picture.jpg doc/examplefolder/
```

- Change the storage class of the file to STANDARD\_IA.

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy examplebucket2-1250000000.ap-beijing.myqcloud.com/doc/picture.jpg
```

```
doc/examplefolder/ -H '{"x-cos-storage-class':'STANDARD_IA'}"
```

- Change the storage class of the file to ARCHIVE and rename it "photo.jpg".

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy
examplebucket2-1250000000.ap-beijing.myqcloud.com/doc/picture.jpg
doc/examplefolder/photo.jpg -H '{"x-cos-storage-class':'Archive'}"
```

## Command syntax for copying a folder

```
coscmd copy -r <sourcepath> <cospath>
```

- **Sample: copying the `examplefolder` directory in the `examplebucket2-1250000000` bucket to the `doc` directory in the `examplebucket1-1250000000` bucket.**

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou copy -r
examplebucket2-1250000000.cos.ap-guangzhou.myqcloud.com/examplefolder
doc/
```

### Note:

- Please replace the parameters in "<>" with the source path (sourcepath) of the file you want to copy from COS and the destination path (cospath) of the file you want to copy to COS.
- The format of sourcepath is: `<BucketName-APPID>.cos.<region>.myqcloud.com/<cos path>` .
- Use the `-d` parameter to set the `x-cos-metadata-directive` header (value range: Copy and Replaced; Copy by default).
- When using the `-H` parameter to set the HTTP header, ensure the format is JSON, for example: `coscmd copy -H -d Replaced '{"x-cos-storage-class':'Archive','Content-Language':'zh-CN'}' <localpath> <cospath>` . For more headers, refer to the [PUT Object - Copy](#) documentation.

## Moving a file or folder

### Note:

The `<sourcepath>` and `<cospath>` of the move command must not be the same, otherwise the file will be deleted. This is because the move command first copies, then

deletes, and the file under the `<sourcepath>` will eventually be removed.

## Command syntax for moving a file

```
coscmd move <sourcepath> <cospath>
```

- **Sample (intra-bucket movement): moving "picture.jpg" in the `examplebucket-1250000000` bucket to the "doc" folder.**

```
coscmd -b examplebucket-1250000000 -r ap-chengdu move examplebucket-1250000000.ap-chengdu.myqcloud.com/picture.jpg doc/
```

- **Sample (cross-bucket movement): moving "picture.jpg" in the `examplebucket2-1250000000` bucket to "doc/folder/" in the `examplebucket1-1250000000` bucket.**

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.ap-beijing.myqcloud.com/picture.jpg doc/folder/
```

- **Sample: changing the storage class of the file to STANDARD\_IA.**

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.ap-beijing.myqcloud.com/picture.jpg doc/folder/ -H '{"x-cos-storage-class':'STANDARD_IA'}"
```

- **Sample: changing the storage class of the file to ARCHIVE.**

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move examplebucket2-1250000000.ap-beijing.myqcloud.com/data/exampleobject data/examplefolder/exampleobject -H '{"x-cos-storage-class':'Archive'}"
```

## Command syntax for moving a folder

```
coscmd move -r <sourcepath> <cospath>
```

- **Sample: moving the "examplefolder" directory in the `examplebucket2-1250000000` bucket to the "doc" directory in the `examplebucket1-1250000000` bucket.**

```
coscmd -b examplebucket1-1250000000 -r ap-guangzhou move -r
examplebucket2-1250000000.cos.ap-guangzhou.myqcloud.com/examplefolder
doc/
```

#### ⓘ Note:

- Replace the parameters in "<>" with the source path (sourcepath) of the file you want to move on COS and the destination path (cospath) on COS.
- The format of sourcepath is: `<BucketName-APPID>.cos.<region>.myqcloud.com/<cos path>` .
- Use the `-d` parameter to set the `x-cos-metadata-directive` header (value range: Copy and Replaced; Copy by default).
- When using the `-H` parameter to set the HTTP header, ensure the format is JSON.  
**Example:** `coscmd move -H -d Replaced '{"x-cos-storage-class':'Archive','Content-Language':'zh-CN'}" <localpath> <cospath>` . For more headers, see the [PUT Object - copy](#) documentation.

## Setting object access permission

- Command format.

```
coscmd putobjectacl --grant-<permissions> <UIN> <cospath>
```

- **Sample: granting `100000000001` permission to read "picture.jpg".**

```
coscmd putobjectacl --grant-read 100000000001 picture.jpg
```

- **Sample: querying the file's access permission.**

```
coscmd getobjectacl picture.jpg
```

## Enabling/Suspending versioning

- Command format.

```
coscmd putbucketversioning <status>
```

- **Sample: enabling versioning.**

```
coscmd putbucketversioning Enabled
```

- **Sample: suspending versioning.**

```
coscmd putbucketversioning Suspended
```

- **Sample: querying versioning.**

```
coscmd getbucketversioning
```

#### **Note:**

- Please replace the parameters within "<>" with the version control status (status) you require.
- Once versioning is enabled for the bucket, it cannot return to the prior status (initial status). However, you can suspend versioning for the bucket so that subsequent uploads of objects will not generate multiple versions.

## Restoring an archived file

- **Command syntax for restoring an archived file.**

```
coscmd restore <cospath>
```

- **Sample: restoring "picture.jpg" using the expedited retrieval mode (effective for 3 days).**

```
coscmd restore -d 3 -t Expedited picture.jpg
```

- **Command syntax for restoring archived files.**

```
coscmd restore -r <cospath>
```

- **Sample:** restoring the "examplefolder/" folder using the expedited retrieval mode (effective for 3 days).

```
coscmd restore -r -d 3 -t Expedited examplefolder/
```

**Note:**

- Please replace the parameters in "<>" with the COS path (cospath) of the files you want to query.
- Use `-d <day>` to set the expiration time for temporary copies. Default value: 7.
- Use `-t <tier>` to specify the retrieval mode, with enumeration values: Expedited (fast retrieval mode), Standard (standard retrieval mode), and Bulk (batch retrieval mode). The default value is Standard.

## FAQs

If you have any questions about COSCMD, see [COSCMD](#).

## Summary

Of course, COS not only offers the above applications and services but also provides a variety of popular open-source applications integrated with Tencent Cloud COS plugins. Click [here](#) to launch and start using them immediately!

# COSCLI (Beta)

## COSCLI Overview

Last updated: 2025-11-19 15:53:31

COS provides the command-line client COSCLI to allow you to upload, download, delete, and perform other operations on COS objects by using simple commands.

COSCLI is written in Go and adopts the Cobra framework. It supports multi-bucket configurations and cross-bucket operations. You can run `./coscli [command] --help` to see how to use COSCLI.

### Feature List

- [Generating and Modifying Configuration Files – config](#)
- [Creating a Bucket – mb](#)
- [Deleting a Bucket – rb](#)
- [Tagging Bucket – bucket-tagging](#)
- [Querying Bucket/Object List – ls](#)
- [Obtaining Statistics on Different Types of Objects – du](#)
- [Uploading/Downloading/Copying Objects – cp](#)
- [Syncing Upload/Download/Copy – sync](#)
- [Deleting Objects – rm](#)
- [Obtaining Object Hash – hash](#)
- [Listing Incomplete Multipart Uploads – lsparts](#)
- [Clearing Incomplete Multipart Uploads – abort](#)
- [Restoring Archived Objects – restore](#)
- [Getting Pre-Signed URLs – signurl](#)
- [Creating/Obtaining a Symbolic Link – symlink](#)
- [Viewing Contents of an Object – cat](#)
- [Listing Contents and Statistics Under a Directory – lsdu](#)
- [Bucket versioning](#)

### Download Address

COSCLI tool supports Windows, MacOS, and Linux operating systems. For details, see [Download and Installation Configuration](#).

# Download and Installation Configuration

Last updated: 2026-02-12 18:24:06

COSCLI provides binary packages for Windows, macOS, and Linux, which can be used after simple installation and configuration.

## Step 1: Download the COSCLI tool

You can choose the download address for the COSCLI tool based on your business scenario. If your server is in China, it's recommended to use the China Site Download Address (The tool versions of the links here are all the latest versions. If you need to use an earlier version, you can visit [release](#) to retrieve historical versions).

China Site Download Address	GitHub Download Address (Recommended for overseas users)	SHA256 Checksum
<a href="#">Windows-386</a>	<a href="#">Windows-386</a>	96e3c2aa2706f2c580f30093d1bf9f468046fd2cf498596f373f8e2707f0dea1
<a href="#">Windows-amd64</a>	<a href="#">Windows-amd64</a>	74f3b5ebbe89be2c013f9d3b2d8968691372801f2671bb31548eab4476fe9179
<a href="#">macOS-amd64</a>	<a href="#">macOS-amd64</a>	432ee022dcfdaec5a77e113386e7bcaa340680ea4c959147a3afb840c664aef9
<a href="#">macOS-arm64</a>	<a href="#">macOS-arm64</a>	df0018fbf78b552cbe875ebe26e8bdf7938c7f4394959f913dfc2ea4d1252568
<a href="#">Linux-386</a>	<a href="#">Linux-386</a>	a58e220bdc14781522b0c9eab76b02e5124b19f57554771a9e7ab204583c89ab
<a href="#">Linux-amd64</a>	<a href="#">Linux-amd64</a>	7165f2ae16c5f7ac495864c963ca574a76e04ec72680d7bc8a8eee3234d8cf91
<a href="#">Linux-arm</a>	<a href="#">Linux-arm</a>	5c5f6b70be061c7c30777dbc67271b2664036fbb56aaf6774c55ccc7f03194ed
<a href="#">Linux-arm64</a>	<a href="#">Linux-arm64</a>	0404b4da5b1d0c230c7d7522cb3bbec2909e314ab998889a0aeb8dc6094a2d21

You can also use the command line to get the COSCLI tool files for macOS and Linux environments from a China site:

- **macOS-amd64:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-da  
rwin-amd64`
- **macOS-arm64:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-da  
rwin-arm64`
- **Linux-386:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-linux-  
386`
- **Linux-amd64:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-lin  
ux-amd64`
- **Linux-arm:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-linux-  
arm`
- **Linux-arm64:** `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-linu  
x-arm64`

**Note:**

The current version on GitHub is v1.0.8. To get the latest version, historical versions, and change logs of the tool, please go to [release](#) to view.

## Step 2: Install the COSCLI Tool

### Windows

1. Take `windows-amd64` version as an example, move the downloaded Windows version of the COSCLI tool to the `C:\Users\< username >` directory.
2. Rename `coscli-windows.exe` to `coscli.exe`.
3. Press `win+r` to open the `Run` program.
4. In the dialog box, enter `cmd` and press `Enter` to open the command line window.
5. In the command prompt, enter the following command.

```
coscli --version
```

If the output is `coscli version v1.0.8`, the installation is successful.

**Note:**

Under the `Windows` system, the method of using COSCLI in different command line clients may vary slightly. If entering `coscli [command]` does not work properly with COSCLI, please try the format `./coscli [command]`.

## macOS

1. Take `macOS-amd64` version as an example, run the following command to rename the macOS version COSCLI file.

```
mv coscli-darwin-amd64 coscli
```

2. Run the following command to modify the file execution permission.

```
chmod 755 coscli
```

3. In the command prompt, enter the following command.

```
./coscli --version
```

If the output is `coscli version v1.0.8`, the installation is successful.

### Note:

When using COSCLI on a macOS, if the prompt `Unable to open "coscli" because the developer cannot be verified` appears, you can go to `Settings > Security and Privacy > General` and select `Still want to open coscli`, afterwards COSCLI can be used normally.

## Linux

1. Take `Linux-amd64` version as an example, run the following command to rename the Linux version COSCLI file.

```
mv coscli-linux-amd64 coscli
```

2. Run the following command to modify the file execution permission.

```
chmod 755 coscli
```

3. In the command prompt, enter the following command.

```
./coscli --version
```

If the output is `coscli version v1.0.8` , the installation is successful.

## Step 3: Configure the COSCLI Tool

### Note:

- It is recommended that users use the tool with a [Temporary Key](#) to further enhance its security through temporary authorization. When applying for a Temporary Key, please follow the [Principle of Least Privilege](#) to prevent leaking resources outside the targeted buckets or objects.
- If you must use a permanent key, it is recommended to follow the [Principle of Least Privilege](#) to limit the permission scope of the permanent key.

When using COSCLI for the first time, you need to initialize the configuration file, which contains the following two parts:

- To authorize COSCLI to access your Tencent Cloud account, you need to configure the secret ID, secret key, and temporary key token.
- To add an alias for a common bucket, you need to configure its name, geographical information, and alias. After configuring this information, users can use the alias for bucket operations without having to fill in the bucket name and geographical information again. Adding configurations for multiple common buckets also facilitates cross-bucket or cross-domain operations more easily. If you don't need to configure common bucket information, you can press `Enter` to skip.

When using COSCLI for the first time, it will automatically call `./coscli config init` to create a configuration file at `~/.cos.yaml` . You can complete the setup interactively through the command line. Later, you can also use `./coscli config init` to interactively generate a configuration file in another location for COSCLI. You can use `./coscli config show` to view the location and configuration parameters of the file.

The configuration items in the configuration file are as described below:

Configuration Item	Required	Description
Secret ID	Yes	<p>Key ID, it is recommended to use a sub-account key and follow the principle of least privilege to reduce risks. Information on obtaining a sub-account key can be found in <a href="#">Sub-account Access Key Management</a>.</p> <div style="border: 1px solid #00a0e3; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> Secret IDs configured via initialization or command configuration are stored in encrypted mode. To manually modify the configuration file, set the Disable Encryption parameter to turn off key encryption.</p> </div>
Secret Key	Yes	<p>Key, it is recommended to use a sub-account key and follow the principle of least privilege to reduce risks. Information on obtaining a sub-account key can be found in <a href="#">Sub-account Access Key Management</a>.</p> <div style="border: 1px solid #00a0e3; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> Secret Keys configured via initialization or command settings are stored in encrypted mode. To manually edit the configuration file, set the Disable Encryption parameter to turn off key encryption.</p> </div>
Session Token	No	<p>Temporary key token, needs to be configured when using a temporary key; otherwise you can skip by pressing <code>Enter</code>. For more information on temporary keys, see <a href="#">Accessing COS Using a Temporary Key</a>.</p> <div style="border: 1px solid #00a0e3; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> Session Tokens configured via initialization or command settings are stored in encrypted mode. To manually edit the configuration file, set the Disable Encryption parameter to turn off key encryption.</p> </div>

Mode	No	Set the identity mode, supporting enumerated values <code>SecretKey</code> and <code>CvmRole</code> . It can be null, with the default null value being <code>SecretKey</code> , indicating the use of a key request COS. When Mode is <code>CvmRole</code> , it means using <a href="#">Managing Instances Roles</a> to request COS.
Cvm Role Name	No	Set the CVM role instance name, see <a href="#">Managing Instances Roles</a> for details.
protocol	No	Network transfer protocol, which is HTTPS by default. If you want to change it to HTTP, directly modify it in the configuration file.
APPID	Yes	APPID is the account you receive after successfully applying for a Tencent Cloud account, automatically assigned by the system, and can be obtained from <a href="#">Account Information</a> . A bucket's full name is composed of two elements: <code>BucketName</code> and <code>APPID</code> , formatted as <code>&lt;BucketName-APPID&gt;</code> , for details please refer to <a href="#">Bucket Naming Conventions</a> .
Bucket Name	Yes	Bucket name, together with APPID, constitutes the bucket's full name, formatted as <code>&lt;BucketName-APPID&gt;</code> , for details please refer to <a href="#">Bucket Naming Conventions</a> .
Bucket Endpoint	Yes	Domain name of the region where the bucket is located, default domain format is <code>cos.&lt;region&gt;.myqcloud.com</code> , where <code>&lt;region&gt;</code> represents the bucket's region, such as <code>ap-guangzhou</code> , <code>ap-beijing</code> , etc. For a list of supported regions by COS, see <a href="#">Regions and Access Endpoints</a> . If the bucket has global acceleration enabled, you can configure a global acceleration domain name. For example, the global acceleration domain name is set to <code>cos.accelerate.myqcloud.com</code> ; the internal network global acceleration domain name is set to <code>cos-internal.accelerate.tencentcos.cn</code> .
Bucket Alias	No	Bucket alias, once configured, you can use <code>BucketAlias</code> to replace <code>BucketName-APPID</code> , reducing the command length required for input. If this option is not configured, the value of <code>BucketAlias</code> is the same as <code>BucketName-APPID</code> .
		Metadata acceleration bucket tag, used to identify whether the bucket has the <a href="#">metadata acceleration</a>

OFS Bucket	No	feature enabled. Currently, the metadata acceleration feature is in public beta and requires you to <a href="#">contact us</a> to apply for use. There is no charge during the public beta period, and you can follow your <a href="#">internal messages</a> or <a href="#">billing overview</a> to learn about the latest billing updates.
CloseAutoSwitchHost	No	Set whether to disable automatic switching to a backup domain. Optional values true   false, can be null. <ul style="list-style-type: none"> <li>If not set or set to false, backup domain switching will be executed;</li> <li>If set to true, backup domain switching will not be executed.</li> </ul>
DisableEncryption	No	Set whether to disable key encryption. Optional value: true   false, which can be left blank. <ul style="list-style-type: none"> <li>If not set or the value is false, the key-related information in the configuration file will be encrypted.</li> <li>Set to true, and the key-related information in the configuration file will not be encrypted.</li> </ul>
DisableAutoFetchBucketType	No	Set whether to automatically retrieve the COS bucket type. Available values: true   false. Can be left blank. <ul style="list-style-type: none"> <li>If not set or set to false, the tool automatically selects the COS bucket type, and authorization <code>cos:HeadBucket</code> is required.</li> <li>If set to true, the tool will not automatically identify the bucket type but preferentially determine it based on the <code>--bucket-type</code> parameter. If this parameter is not set, the bucket type is determined by the <code>ofs</code> parameter in the configuration file (true means an OFS bucket, false means a COS bucket). If the configuration file does not include bucket information, the COS bucket type is used by default.</li> </ul>

During the initial configuration, COSCLI will ask you to configure information for just one bucket. If you want to configure multiple buckets, you can later use the `./coscli config add` command to add bucket configurations. If you need to modify the configuration file or learn more about operations related to the configuration file, refer to the [config command](#) or use the `./coscli config --help` command for quick access to related instructions. Before starting to use commands, you can quickly check how to use COSCLI by running the command `./coscli --help`.

## Other configuration methods

In addition to generating the configuration file interactively with the `./coscli config init` command, you can also manually write the COSCLI configuration file. The format of the COSCLI configuration file is in `yaml` format, with an example configuration as follows:

```
cos:
  base:
    secretid: XXXXXXXXXXXXXXXXXXXX
    secretkey: XXXXXXXXXXXXXXXXXXXX
    sessiontoken: ""
    protocol: https
  buckets:
  - name: examplebucket1-1250000000
    alias: bucket1
    region: ap-shanghai
    endpoint: cos.ap-shanghai.myqcloud.com
    ofs: false
  - name: examplebucket2-1250000000
    alias: bucket2
    region: ap-guangzhou
    endpoint: cos.ap-guangzhou.myqcloud.com
    ofs: false
  - name: examplebucket3-1250000000
    alias: bucket3
    region: ap-chengdu
    endpoint: cos.ap-chengdu.myqcloud.com
    ofs: false
```

**Note:**

COSCLI by default reads configuration items from `~/.cos.yaml`. If users want to use their own defined configuration file, please use the `-c` (`--config-path`) option in the command. The configuration for CFS's `secretid/secretkey/sessiontoken` are all encrypted strings.

# Common Options

Last updated: 2025-11-07 17:22:04

You can view the general options supported by COSCLI using the `./coscli --help` or `./coscli -h` command.

## Option Description

The following are common options for COSCLI, which can be used in all its commands:

### Note:

- We recommend using the tool with [temporary keys](#) to enhance security through temporary authorization. When applying for temporary keys, please follow the [principle of least privilege](#) to prevent leakage of resources beyond the target bucket or object.
- If you must use a permanent key, we recommend you follow the [Notes on Principle of Least Privilege](#) to limit the scope of permission on the permanent key.

Option	Note
<code>-h, --help</code>	Outputs help information. You can view the help information and usage of the tool with the <code>-h</code> or <code>--help</code> command. You can also enter <code>-h</code> after each command (with no parameter appended) to see how to use the command. For example, to view the specific usage of the bucket creation command, enter <code>coscli mb -h</code> .
<code>-c, --config-path</code>	Configuration file path, which is <code>~/.cos.yaml</code> for COSCLI by default. You can also specify a custom configuration file by adding <code>-c</code> after a command.
<code>-e, --endpoint</code>	In addition to configuring the region of a bucket in advance in the configuration file, you can also use <code>-e</code> in COSCLI to specify the bucket endpoint in the format of <code>cos.&lt;region&gt;.myqcloud.com</code> , where <code>&lt;region&gt;</code> represents the bucket region, such as <code>ap-guangzhou</code> and <code>ap-beijing</code> . For the list of regions supported by COS, see <a href="#">Regions and Access Endpoints</a> .
<code>-i, --secret-id</code>	Specifies the <code>SecretId</code> used to access COS.
<code>-k, --secret-key</code>	Specifies the <code>SecretKey</code> used to access COS.
<code>--session-token</code>	Access COS with a temporary key.
<code>-v, --version</code>	Displays the COSCLI version.
<code>-p, --protocol</code>	Network transfer protocol, which is HTTPS by default.
<code>--init-skip</code>	By default, it is false. If it is set to true ( <code>--init-skip=true</code> ), skip the config init interactive operation, and directly use the <code>SecretId</code> , <code>SecretKey</code> , and endpoint parameters to request the APIs. When this parameter is used, the <code>-i</code> , <code>-k</code> , and <code>-e</code> parameters must be configured.
<code>--log-path</code>	Customize the <code>coscli.log</code> file location, in the same directory as COSCLI by default. Specify a directory or specific file (the file must end with <code>.log</code> ), for example: <code>/data/</code> or <code>/data/coscli.log</code> .
<code>--customize</code>	By default, it is false. When set to true, enable the custom domain name

customize d	and specify it using the <code>--endpoint (-e)</code> parameter.
<code>-- disable- log</code>	Defaults to <code>false</code> . When set to <code>true</code> , it disables the generation of <code>coscli.log</code> and the output of corresponding logs.
<code>--bucket- type</code>	Specify the current bucket access type (COS/OFS).

## Sample

### Example 1: Switching bucket to upload an object

When you need to switch to a bucket in another region through COSCLI, you can use the `-e` option to specify the endpoint of the bucket.

For example, to upload the local file `test.txt` to the bucket `examplebucket-1250000000` in the Chengdu region with the endpoint `cos.ap-chengdu.myqcloud.com`, run the following command:

```
./coscli cp test.txt cos://examplebucket-1250000000/test.txt -e cos.ap-
chengdu.myqcloud.com
```

### Example 2: Switching user account to view the file list

When you need to use the identity of another account, you can use the `-i` and `-k` options to specify the `SecretId` and `SecretKey` of your key respectively.

For example, to use the identity of another account to list the files in the bucket `examplebucket-1250000000` in the Chengdu region, run the following command:

```
./coscli ls cos://examplebucket-1250000000 -e cos.ap-
chengdu.myqcloud.com -i ***** -k
*****
```

# Common Commands

## Generating and Modifying Configuration Files – config

Last updated: 2025-11-07 17:23:09

### Command format

The `config` command is used to generate and modify the configuration file:

```
./coscli config [command] [flag]
```

#### Note:

- If you have set the configuration items correctly, you can run `./coscli config show` to view the configurations.
- For more general options for this command (such as switching buckets or user accounts), see [General Options](#).
- The generated configuration file uses the HTTPS protocol by default. If you want to change it to HTTP, directly modify it in the configuration file.

`config` includes the following sub-commands:

Command	Description
<code>add</code>	Adds a new bucket configuration.
<code>delete</code>	Deletes an existing bucket configuration.
<code>init</code>	Generates the configuration file interactively.
<code>set</code>	Modify one or more configuration items in the base group of the configuration file. The base group includes <code>secretid</code> , <code>secretkey</code> , <code>sessiontoken</code> , <code>mode</code> , and <code>cvmrolename</code> information.
<code>show</code>	Prints information in a specific configuration file.

`config` and its sub-commands include the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	View the detailed usage of this command.
-c	--config-path	Path of the configuration file to use.

The `config add` sub-command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	View the detailed usage of this command.
-a	--alias	Bucket alias.
-b	--bucket	Bucket name.
-r	--region	Region of the bucket.
-o	--ofs	Metadata acceleration bucket flag. For more information, see <a href="#">Metadata Acceleration Overview</a> .

**Note:**

If you need to specify the bucket endpoint, you can use the general flag `-e` or `--endpoint`. For more information, see [General Options Introduction](#).

The `config delete` sub-command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	View the detailed usage of this command.
-a	--alias	Bucket alias.

The `config set` sub-command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	View the detailed usage of this command.
None	--secret_id	Sets the secret ID, which can be created and obtained from the <a href="#">CAM console</a> .
None	--secret_key	Sets the secret key, which can be created and obtained from the <a href="#">CAM console</a> .
-t	--session_token	Sets the temporary key token. For more information on temporary key, see <a href="#">Accessing COS Using a Temporary Key</a> .
None	--mode	Set the identity mode, supporting enumeration values <code>SecretKey</code> and <code>CvmRole</code> . It can be empty, with a default value of <code>SecretKey</code> , which means using a key to request COS. When the mode is <code>CvmRole</code> , it means using <a href="#">manage instance role</a> to request COS.
None	--cvm_role_name	Set the CVM role instance name. For details, see <a href="#">manage instance role</a> .
None	--close_auto_switch_host	Set whether to close automatic switching of the backup domain. Available values: true, false. Can be left blank. <ul style="list-style-type: none"> <li>If not set or the value is false, the domain name will be switched.</li> <li>If set to true, the domain name will not be switched.</li> </ul>
None	--disable_encryption	Set to turn off key encryption. It defaults to false. If set to true, the key-related information in the configuration file will be unencrypted.
None	--disable_auto_fetch_bucket_type	Set whether to disable automatically obtaining the bucket type. Available values: true, false. <ul style="list-style-type: none"> <li>If not set or set to false, the bucket type will be obtained automatically (ensure the account used has <code>cos:HeadBucket</code> permission).</li> <li>If set to true, the bucket type will not be obtained automatically and the bucket type in the user profile will be used (for example, if the Bucket</li> </ul>

Configuration item ofs in the configuration file is set to true, the bucket type is ofs; if set to false, the bucket type is cos).

## Sample

### Adding a new bucket configuration

```
./coscli config add -b examplebucket3-1250000000 -r ap-chengdu -e  
cos.ap-chengdu.myqcloud.com -a bucket3
```

### Deleting an existing bucket configuration

```
./coscli config delete -a bucket3
```

### Modifying `session-token` in the default configuration file

```
./coscli config set --session_token test-token123
```

### Printing information in a specific configuration file

```
./coscli config show -c /your/config/path.yaml
```

### Modifying `mode` and `cvmrolename` in the default configuration file

```
./coscli config set --mode CvmRole --cvm_role_name testName
```

# Creating Buckets – mb

Last updated: 2025-11-12 15:45:48

The `mb` command is used to create a bucket.

## Note:

If you need to use this command, when you perform [authorization policy](#), set action to `cos:PutBucket`. For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli mb cos://<BucketName-APPID> -e <endpoint> [flag]
```

## Note:

To create a bucket using the `mb` command, you need to carry the global flag `-e` or `--endpoint` to specify the region where the bucket is located.

`mb` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;BucketName-APPID&gt;</code>	Customizes the bucket name	<code>cos://examplebucket-1250000000</code>

`mb` includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
-r	--region	Region of the bucket.
-m	--maz	Create a multi-az bucket
-o	--ofs	Create a bucket for ofs
None	--acl	Set the ACL of the bucket, such as private, public-read, and public-read-write.
None	--grant-read	Grant the authorized entity permission to read the bucket. The format is id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-read-acp	Grant the authorized entity permission to read the bucket's access control list (ACL). The format is id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-write-acp	Grant the authorized entity permission to write to the bucket's access control list (ACL). The format is id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-full-control	Grant the authorized entity all permissions to operate the bucket. The format is id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--tags	Bucket tag collection, with a maximum of 50 tags (for example, --tags="Key1=Value1&Key2=Value2")

 **Note:**

- After you run the `mb` command to successfully create a bucket, we recommend that you add information about the bucket in the configuration file, so that you can use the bucket alias for quick operations. For command usage, see the following example.
- For more general options for this command (such as switching buckets or user accounts), see [General Options](#).

## Sample: Create the bucket3-1250000000

```
./coscli mb cos://bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com
```

If you want to configure an alias for the bucket you just created, update the configuration file with the following command:

```
./coscli config add -b bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com  
-a bucket3
```

After the update, you can access the bucket at `cos://bucket3`.

# Deleting Buckets – rb

Last updated: 2025-05-12 15:06:28

The `rb` command is used to delete a bucket.

## Note:

If you need to use this command, when you perform [authorization policy](#), set action to `cos:DeleteBucket`. For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli rb cos://<bucket-name> [flag]
```

`rb` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;bucket-name&gt;</code>	Specify the bucket to access. You can use the bucket alias from the <a href="#">configuration parameters</a> or the bucket name. If using the bucket name, you must also include the <code>endpoint</code> flag.	<ul style="list-style-type: none"> <li>Access using bucket alias: <code>cos://example-alias</code></li> <li>Access using bucket name: <code>cos://examplebucket-1250000000</code></li> </ul>

`rb` includes the following optional flags:

Flag Abbreviation	Flag Name	Description
<code>-h</code>	<code>--help</code>	Views the usage of this command.
<code>-r</code>	<code>--region</code>	Indicates the region of the bucket.

## Note:

- The `rb` command can only delete empty buckets. If your bucket still contains files, please use the [Delete Files](#) and [Clean Fragments](#) commands to clear the files and file fragments in the bucket before deleting it.
- After you run the `rb` command to delete the bucket successfully, we recommend that you delete the bucket information in the configuration file. For command usage,

see the following example.

- For more general options for this command (such as switching buckets or user accounts), see [General Options](#).

## Sample

```
// Delete the bucket3 bucket.(require confirmation operation)
./coscli rb cos://bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com
```

```
// Update configuration file, delete bucket configuration of bucket3.
./coscli config delete -a bucket3
```

# Tagging Bucket – bucket-tagging

Last updated: 2025-11-12 15:49:33

The `bucket-tagging` command is used to create (modify), get, and delete bucket tags. One bucket can have up to 50 tags.

## Note:

- If you need to obtain bucket tags, when you perform [authorization policy](#), set action to `cos:GetBucketTagging`.
- If you need to set the bucket tags, when you perform [authorization policy](#), set action to `cos:PutBucketTagging`.
- If you need to delete bucket tags, when you perform [authorization policy](#), set action to `cos>DeleteBucketTagging`.

For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli bucket-tagging --method [method] cos://<bucket-name> [tag_key]#
[tag_value]
```

`bucket-tagging` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;bucket-name&gt;</code>	Specify the bucket to access. You can use the bucket alias from the <a href="#">configuration parameters</a> or the bucket name. If using the bucket name, you must also include the <code>endp oint</code> flag.	Access using bucket alias: <code>cos://example-alias</code> Access using bucket name: <code>cos://examplebucket-1250000000</code>

The `bucket-tagging` command contains the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	View the usage of this command.
None	--method	Specify the required operations, including put (set tag), get (query tag), delete (delete tag), and add (add tag).

**Note:**

For other general options related to this command (such as switching buckets or user accounts), see the [General Options](#) documentation.

## Setting Bucket Tags

Bucket tags are represented by a set of Key-Value pairs. Only the bucket owner and users with PutBucketTagging permissions can set bucket tags, otherwise it will return error code `403 AccessDenied`.

### Command format

```
./coscli bucket-tagging --method put cos://bucketAlias key1#value1
key2#value2
```

Among them, `key#value` represents the tag key-value pair Key-Value, where key and value are separated by `#`. If the bucket does not have a tag, this command will add the specified tag to the bucket; otherwise, it will overwrite the original tag.

### Sample

Configure two group tags for the bucket named example-alias. One tag has key 1 and value 111, the other has key 2 and value 222. The command is as follows:

```
./coscli bucket-tagging --method put cos://example-alias 1#111 2#222
```

## Querying bucket tags

### Command format

```
./coscli bucket-tagging --method get cos://bucketAlias
```

## Sample

```
./coscli bucket-tagging --method get cos://example-alias
```

The output below indicates that the bucket with the alias `example-alias` has two sets of tags configured. One set has a key of 1 and a value of 111, while the other set has a key of 2 and a value of 222.

```
KEY | VALUE
-----+-----
 1 | 111
 2 | 222
```

## Deleting All Bucket Tags

### Command format

```
./coscli bucket-tagging --method delete cos://bucketAlias
```

## Sample

```
./coscli bucket-tagging --method delete cos://example-alias
```

## Adding Bucket Tags

Bucket tags are represented by a set of Key-Value pairs. Only the bucket owner and users with PutBucketTagging permissions can add bucket tags, otherwise it will return error code `403 AccessDenied`.

### Command Syntax

```
./coscli bucket-tagging --method add cos://bucketAlias key1#value1
key2#value2
```

Among them, `key#value` represents the tag key-value pair Key-Value, where key and value are separated by `#`.

## Operation Example

Add a group tag to the bucket named `example-alias`. The tag has key `1` and value `111`. The command is as follows:

```
./coscli bucket-tagging --method add cos://example-alias 1#111
```

# Querying Bucket/Object List – ls

Last updated: 2026-02-12 18:20:03

The `ls` command is used to query the list of buckets, objects in a bucket, and objects in a directory.

## Note:

- If you need to list files in the bucket, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucket`.
- If you need to list historical version information (import `--all-versions`), when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucketVersioning`, `cos:GetBucketObjectVersions`.
- If you need to list buckets under the account, when you perform [authorization policy](#), set action to `cos:GetService`.

For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli ls [cos://<bucket-name>[/prefix/]] [flag]
```

`ls` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;bucket-name&gt;</code>	Optional parameter. Specifies the bucket to access. Supports using the bucket alias from <a href="#">configuration parameters</a> or the bucket name. If using the bucket name, the <code>endpoint</code> flag must be provided additionally.	Access using bucket alias: <code>cos://example-alias</code> Access using bucket name: <code>cos://examplebucket-1250000000</code>
<code>/prefix/</code>	Specifies a directory (optional).	<code>/picture/</code>

`ls` includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--include	Includes specific objects.
None	--exclude	Excludes specific objects.
-r	--recursive	Specifies whether to traverse directories recursively and list all objects.
None	--limit	Specifies the maximum quantity to be listed. (If not specified or set to 0, the default limit is 10,000 files. When set to -1, all COS files can be retrieved)
None	--all-versions	List all versions of objects. It is available only after version control is enabled for a bucket. Add display fields <code>VersionId</code> , <code>IsLatest</code> and <code>Delete Marker</code> for parameters when listing historical versions.

### Note

- `--include` and `--exclude` support standard regular expressions. You can use regular expressions to filter objects that meet your requirements.
- When using `zsh`, you may need to enclose the pattern string with double quotation marks.

```
./coscli ls cos://bucket1 -r --include ".*\.mp4$"
```

- For more general options for this command (such as switching buckets or user accounts), see [General Options](#).

## Sample

### Listing all buckets of the current account

```
./coscli ls
```

The returned information includes the bucket name, region, creation time, and total number of buckets. Below is an example:

```

      BUCKET NAME      |      REGION      |      CREATE DATE
-----+-----+-----
--
examplebucket-1250000000 | ap-nanjing      | 2022-01-01T00:00:00Z
-----+-----+-----
--
                                TOTAL BUCKETS: |          2
-----+-----
--

```

## Listing objects

### Listing all objects in the `bucket1` bucket

```
./coscli ls cos://bucket1
```

The returned information includes the object key (the unique identifier of the object in the bucket), storage class, last update time, object size, and total number of objects. Below is an example:

```

      KEY      |      TYPE      |      LAST MODIFIED      |
ETAG          |      SIZE      |      RESTORESTATUS
-----+-----+-----+-----
--
test.txt     | STANDARD      | 2024-06-05T15:03:37+08:00 |
"a3bc6c9058109f8da48d41a5ab9abc7c" | 4.88 KB      |
-----+-----+-----+-----
--
TOTAL OBJECTS: |          1
-----+-----
--

```

### Listing all objects and subdirectories in the `picture` directory in the `bucket1` bucket

```
./coscli ls cos://bucket1/picture/
```

General listing only returns the data at the level of the query path, without expanding subpaths. Below is an example:

```

      KEY                |   TYPE   |   LAST MODIFIED   |
ETAG                |   SIZE   | RESTORESTATUS
-----+-----+-----+-----
picture/a4431470f55662 | STANDARD | 2024-06-05T15:03:58+08:00 |
"ed0430c5f27e76605e0555c260478112" | 358.00 B |
picture/e98c6cefa4abd6 | STANDARD | 2024-06-05T15:03:58+08:00 |
"bd5a4bd7248e7dfdb796383bee60470b" | 53.00 B |
-----+-----+-----+-----

TOTAL OBJECTS: |          3
-----+-----+-----+-----

```

### Listing all objects in the `picture` directory in the `bucket1` bucket recursively

```
./coscli ls cos://bucket1/picture/ -r
```

If there are subpaths at the level of the query path, recursive listing will scan all subpaths and return all files under the level of the query path. Below is an example:

```

      KEY                |   TYPE   |   LAST MODIFIED   |
ETAG                |   SIZE   | RESTORESTATUS
-----+-----+-----+-----
picture/subfolder    | DIR      |
|
picture/subfolder/pic2.png | STANDARD | 2024-06-05T15:03:58+08:00 |
"bd5a4bd7248e7dfdb796383bee60470b" | 53.00 B |
-----+-----+-----+-----

TOTAL OBJECTS: |          3
-----+-----+-----+-----

```

```
-----+-----
```

## Listing all MP4 objects in the `bucket1` bucket recursively

```
./coscli ls cos://bucket1 -r --include ".*\.mp4$"
```

## Listing all non-MP4 objects in the `bucket1` bucket recursively

```
./coscli ls cos://bucket1 -r --exclude ".*\.mp4$"
```

## Listing all non-JPG objects prefixed with `test` in the `picture` directory in the `bucket1` bucket

```
./coscli ls cos://bucket1/picture -r --include "^picture/test.*" --  
exclude ".*\.jpg$"
```

## List Historical Versions

### List All Earlier Versions in `bucket1`

```
./coscli ls cos://bucket1/ -r --all-versions
```

If there are subpaths at the level where the query path is located, recursive listing will scan all subpaths and return all historical versions of all files under the level of the query path. An example is as follows:

```

      KEY                |   TYPE   |   VERSIONID
| ISLATEST | DELETE MARKER |   LAST MODIFIED   |
ETAG                |   SIZE
-----+-----+-----+-----
-+-----+-----+-----+-----
-----+-----
cmd/cmd/abort.go    | STANDARD | MTg0NDUwMDM1MjIxMDM3MDI1MDM
| false   | false   | 2025-02-26T14:33:25+08:00 |
"c9bfc40db6669e9a7aee03abcd8b66e8" | 1.89 KB
cmd/cmd/abort_test.go | STANDARD | MTg0NDUwMDM1MjIxMDM3MjgxODI
| false   | false   | 2025-02-26T14:33:25+08:00 |

```

```

"52166b1c60e4089a4652546c0350d2c7" | 4.89 KB
  cmd/cmd/bucket_tagging.go      | STANDARD | MTg0NDUwMDM1MjIxMDM3MjY3MDI
| false      | false      | 2025-02-26T14:33:25+08:00 |
"edad62e08bf65a5bff81304e2b40ac1a" | 3.82 KB
  cmd/cmd/bucket_versioning.go  | STANDARD | MTg0NDUwMDM1MjIxMDM2MTkzMzU
| false      | false      | 2025-02-26T14:33:25+08:00 |
"50a6b0e2e218c437ccdabe2c762aaef1" | 1.86 KB
  cmd/cmd/buket_tagging_test.go | STANDARD | MTg0NDUwMDM1MjIxMDM2MTY4MDE
| false      | false      | 2025-02-26T14:33:25+08:00 |
"cd11257b22c9816df105da15f3ceb70f" | 8.23 KB
  cmd/cmd/abort.go              |          | MTg0NDUwMDM1MjIwNjcyNTcxMDA
| true       | true       | 2025-02-26T14:34:02+08:00 |
|
  cmd/cmd/abort_test.go         |          | MTg0NDUwMDM1MjIwNjc5NzE5NDY
| true       | true       | 2025-02-26T14:34:02+08:00 |
|
  cmd/cmd/bucket_tagging.go     |          | MTg0NDUwMDM1MjIwNjcyNjc5NzI
| true       | true       | 2025-02-26T14:34:02+08:00 |
|
  cmd/cmd/bucket_versioning.go  |          | MTg0NDUwMDM1MjIwNjY3ODc4NzM
| true       | true       | 2025-02-26T14:34:02+08:00 |
|
  cmd/cmd/buket_tagging_test.go |          | MTg0NDUwMDM1MjIwNjY3ODYyMDI
| true       | true       | 2025-02-26T14:34:02+08:00 |
|
-----+-----+-----
-+-----+-----+-----+-----+-----
-----+-----
TOTAL OBJECTS:          | 10
-----+-----

```

# Obtaining Statistics on Different Types of Objects – du

Last updated: 2025-05-12 15:08:27

The `du` command is used to list the statistical information of files per storage type under a bucket or a folder. The statistical information includes the total files of different storage types and the total size of the file per type.

## ⚠ Note:

- If you need to use this command to stat information of all objects, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucket`.
- If you need to use this command to stat all historical version information, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucketVersioning`, `cos:GetBucketObjectVersions`.

For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli du cos://<bucket-name>[/prefix/] [flag]
```

`du` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;bucket-name&gt;</code>	Specify the bucket to access. You can use the bucket alias from the <a href="#">configuration parameters</a> or the bucket name. If using the bucket name, you must also include the <code>endp</code> <code>oint</code> flag.	Access using bucket alias: <code>cos://example-alias</code> Access using bucket name: <code>cos://examplebucket-12500000000</code>
<code>/prefix/</code>	Specifies a directory (optional).	<code>/picture/</code>

`du` includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--include	Includes specific objects.
None	--exclude	Excludes specific objects.
None	--all-versions	All version data of the statistical object is only available after version control is enabled for the bucket. The statistical information will include the count of DeleteMarkers.

### Note

- `--include` and `--exclude` support standard regular expressions. You can use regular expressions to filter objects that meet your requirements.
- When using `zsh`, you may need to enclose the pattern string with double quotation marks.

```
./coscli du cos://bucket1/picture/ --include ".*\.mp4$"
```

- For more general options for this command (such as switching buckets or user accounts), see [General Options](#).

## Sample

### Listing statistics on objects in the `bucket1` bucket

```
./coscli du cos://bucket1
```

The returned information includes the number and size of objects in different storage classes, total number of objects, and total object size in the bucket. Below is an example:

```

STORAGE CLASS      | OBJECTS COUNT | TOTAL SIZE
-----+-----+-----
          STANDARD |             2 |      164 B
          STANDARD_IA |            0 |           0 B
          INTELLIGENT_TIERING |            0 |           0 B

```

```

          ARCHIVE |                0 |                0 B
        DEEP_ARCHIVE |                0 |                0 B
          MAZ_STANDARD |                0 |                0 B
        MAZ_STANDARD_IA |                0 |                0 B
    MAZ_INTELLIGENT_TIERING |                0 |                0 B
          MAZ_ARCHIVE |                0 |                0 B
-----+-----+-----

```

```
INFO[2022-12-14 17:35:41] Total Objects Count: 2
```

```
INFO[2022-12-14 17:35:41] Total Objects Size: 164 B
```

## Listing statistics on objects in the `picture` directory in the `bucket1` bucket

```
./coscli du cos://bucket1/picture/
```

## Listing statistics on all MP4 objects in the `picture` directory in the `bucket1` bucket

```
./coscli du cos://bucket1/picture/ --include ".*\.mp4$"
```

## Listing statistics on all non-MD objects in the `picture` directory in the `bucket1` bucket

```
./coscli du cos://bucket1/picture/ --exclude ".*\.md$"
```

## List the Files in the `bucket1` bucket and the Statistical Information of All Earlier Versions

```
./coscli du cos://bucket1 --all-versions
```

The returned result is as follows. The output information includes: the number and size of objects under each storage type in the bucket, the total number of objects in the bucket, and the total capacity of objects in the bucket.

```

    STORAGE CLASS | OBJECTS COUNT | TOTAL SIZE
-----+-----+-----
          STANDARD |                545 |          1.14 MB

```

```
STANDARD_IA | 0 | 0 B
INTELLIGENT_TIERING | 0 | 0 B
ARCHIVE | 0 | 0 B
DEEP_ARCHIVE | 13 | 11.74 KB
MAZ_STANDARD | 0 | 0 B
MAZ_STANDARD_IA | 0 | 0 B
MAZ_INTELLIGENT_TIERING | 0 | 0 B
MAZ_ARCHIVE | 0 | 0 B
```

```
-----+-----+-----
INFO[2025-02-25 17:36:36] Total Objects Count: 558
INFO[2025-02-25 17:36:36] Total Objects Size: 1.15 MB
INFO[2025-02-25 17:36:36] Total DeleteMarker Count: 501
```

# Uploading/Downloading/Copying Objects – cp

Last updated: 2026-02-26 17:15:29

The `cp` command is used to upload, download, or copy objects.

## Note:

- If you need to use the upload file command, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucket`, `cos:HeadObject`, `cos:InitiateMultipartUpload`, `cos:UploadPart`, `cos:CompleteMultipartUpload`, `cos:ListMultipartUploads`, `cos:ListParts`.
- If you need to use the download file command, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucket`, `cos:HeadObject`, `cos:GetObject`.
- If you need to use the copy file command, when you perform [authorization policy](#), set the target object action to `cos:GetBucket`, `cos:HeadObject`, `cos:InitiateMultipartUpload`, `cos:PutObject`, `cos:CompleteMultipartUpload`. Set the source object action to `cos:HeadBucket`, `cos:GetBucket`, `cos:HeadObject`, `cos:GetObject`.

For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli cp <source_path> <destination_path> [flags]
```

`cp` includes the following parameters:

Parameter Format	Description	Sample
source_path	Source file path, which can be a local path or a COS file path. The COS path is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the <code>endpoint</code> flag.	<ul style="list-style-type: none"> <li>Local path: <code>~/example.txt</code></li> <li>COS file path specified with the bucket alias: <code>cos://bucketalias/example.txt</code></li> <li>COS file path specified with the bucket name: <code>cos://examplebucket-1250000000/example.txt</code></li> </ul>
destination_path	Destination file path, which can be a local path or a COS file path. The COS path is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the <code>endpoint</code> flag.	<ul style="list-style-type: none"> <li>Local path: <code>~/example.txt</code></li> <li>COS file path specified with the bucket alias: <code>cos://bucketalias/example.txt</code></li> <li>COS file path specified with the bucket name: <code>cos://examplebucket-1250000000/example.txt</code></li> </ul>

`cp` includes the following optional flags:

Flag Abbreviation	Flag Name	Description
None	<code>--include</code>	<ul style="list-style-type: none"> <li>Includes specific objects. (Versions prior to v1.0.4 only filter the local file name during upload, while versions v1.0.4 and later will filter the full path.)</li> <li>For example: upload all files under <code>./test</code> to COS. The <code>./test</code> folder contains the <code>aaa</code> folder, which includes the <code>1.txt</code> file. <ul style="list-style-type: none"> <li>versions prior to 1.0.4 will match <code>aaa/1.txt</code></li> <li>1.0.4 and later versions will match <code>./test/aaa/1.txt</code></li> </ul> </li> </ul>
None	<code>--exclude</code>	<ul style="list-style-type: none"> <li>Excludes specific objects. (Versions prior to v1.0.4 only filter the local file name during upload, while versions v1.0.4 and later will filter the full path.)</li> <li>For example: upload all files under <code>./test</code> to COS. The <code>./test</code> folder contains the <code>aaa</code> folder, which includes the <code>1.txt</code> file. <ul style="list-style-type: none"> <li>versions prior to 1.0.4 will match <code>aaa/1.txt</code></li> <li>1.0.4 and later versions will match <code>./test/aaa/1.txt</code></li> </ul> </li> </ul>
<code>-r</code>	<code>--recursive</code>	Specifies whether to traverse all objects in the directory recursively.
None	<code>--storage-class</code>	Specifies the storage class of the uploaded file. Default value: <code>STANDARD</code> . For more information, see <a href="#">Storage Class Overview</a> .
None	<code>--part-size</code>	The part size of the file (default 32 MB, supports up to 5 GB). If you need to adaptively adjust the part size based on file size, set it to 0.
None	<code>--thread-num</code>	Number of concurrent threads. Default value: 5.
None	<code>--rate-limiting</code>	Speed limit for a single URL in MB/s. Value range: 0.1-100 MB/s.
None	<code>--meta</code>	Metadata of the uploaded file, including certain HTTP standard attributes (HTTP Header) and custom metadata prefixed with <code>x-cos-meta-</code> (User Meta). The file metadata is in the format of <code>header:value#header:value</code> , such as <code>E xpires:2022-10-12T00:00:00.000Z#Cache-Control:no-cache#Content-Encoding:gzip#x-cos-meta-x:x</code> .

None	<code>--routines</code>	Specifies the number of files for concurrent upload or download of threads between files, with the default number being <code>3</code> .
None	<code>--fail-output</code>	This option determines whether to enable error output of files when uploads or downloads fail (when the default value is <code>true</code> , it is enabled). If it is enabled, failed file transfers will be recorded in the specified directory (if no directory is specified, the default one is <code>./coscli_output</code> ). If it is disabled, only the number of failed files will be output to the console.
None	<code>--fail-output-path</code>	This option is used to specify the error output folder for recording failed uploads or downloads. By providing a custom folder path, you can control the location and name of the error output folder. If this option is not set, the default error log folder <code>./coscli_output</code> will be used.
None	<code>--retry-num</code>	Number of frequency limit retry times (default value is <code>0</code> ; No retry). <code>1-10</code> times can be selected. When multiple machines execute download operations at the same COS directory simultaneously, you can specify this parameter to retry and avoid frequency limit errors.
None	<code>--err-retry-num</code>	Number of error retry times (default value is <code>0</code> ). <code>1-10</code> times are specified, or if the value is set to <code>0</code> , no retry is performed.
None	<code>--err-retry-interval</code>	Retry interval (only available when <code>--err-retry-num</code> is specified as <code>1-10</code> ). Specify a retry interval of <code>1-10</code> seconds. If it is not specified or set to <code>0</code> , the value of each retry interval will be random among <code>1-10</code> seconds.
None	<code>--only-current-dir</code>	Whether to only upload files in the current directory; ignore subdirectories and their content ( <code>false</code> is set by default; not ignored).
None	<code>--disable-all-symlink</code>	Whether to ignore the subfiles and subdirectories of all the soft links during upload ( <code>true</code> is set by default; not uploaded). Currently only supported on Linux and MacOS systems
None	<code>--enable-symlink-</code>	Whether to upload subdirectories of soft links ( <code>false</code> is set by default; not uploaded). Currently only supported on

	<code>dir</code>	Linux and MacOS systems
None	<code>--disable-crc64</code>	Whether to disable the CRC64 data validation ( <code>false</code> is set by default; validation enabled).
None	<code>--disable-checksum</code>	The default is true, verifying only the fragment crc64. When set to false, it verifies the entire file crc64. (coscli V1.0.6 and prior versions default to false)
None	<code>--move</code>	The source file will be deleted after the file is successfully copied to the target path (only available between cos paths).
None	<code>--version-id</code>	Download a specified version file, which is only supported in a Bucket with version control enabled (single file only).
None	<code>--process-log</code>	Whether process log is enabled, the default is true, enabled
None	<code>--process-log-path</code>	This option is used to specify a dedicated output folder to store process logs. The logs will record information related to file upload or download, including error logs, normal execution logs, retry details. By providing a custom folder path, you can control the location and name of the log output folder. If not set, the default log folder (coscli_output) will be used.
None	<code>--skip-dir</code>	Defaults to false. When set to true, it skips folders during transmission.
None	<code>--acl</code>	Set the file's ACL, such as private, public-read.
None	<code>--grant-read</code>	Grant the authorized entity permission to read the object. The format is <code>id="[OwnerUin]"</code> , for example <code>id="100000000001"</code> . Use comma (half-width) to separate multiple authorized entities, such as <code>id="100000000001",id="100000000002"</code> .
None	<code>--grant-read-acp</code>	Grant the authorized entity permission to read the object's access control list (ACL). The format is <code>id="[OwnerUin]"</code> , for example <code>id="100000000001"</code> . Use comma (half-width) to separate multiple authorized entities, such as <code>id="100000000001",id="100000000002"</code> .
..	<code>--grant-</code>	Grant the authorized entity permission to write to the object's access control list (ACL). The format is <code>id="</code>

None	<code>--write-acp</code>	[OwnerUin]", for example <code>id="100000000001"</code> . Use comma (half-width) to separate multiple authorized entities, such as <code>id="100000000001",id="100000000002"</code> .
None	<code>--grant-full-control</code>	Grant the authorized entity all privileges on the operation object. The format is <code>id="[OwnerUin]"</code> , for example <code>id="100000000001"</code> . Use comma (half-width) to separate multiple authorized entities, such as <code>id="100000000001",id="100000000002"</code> .
None	<code>--tags</code>	The object tag collection supports up to 10 tags (for example, <code>--tags="Key1=Value1&amp;Key2=Value2"</code> )
None	<code>--forbid-overwrite</code>	<p>For buckets with Versioning Not Enabled, when uploading files, specify whether to forbid overwriting objects with the same name.</p> <ul style="list-style-type: none"> <li>• When set to false, default overwrite for objects with the same name.</li> <li>• When set to true, it means overwriting an Object with the same name is forbidden.</li> </ul> <p>When the bucket is in enabled or paused versioning status, the <code>x-cos-forbid-overwrite</code> header setting is invalid, allowing overwriting objects with the same name.</p>
None	<code>--encryption-type</code>	Server-side encryption method (SSE-COS/SSE-C).
None	<code>--server-side-encryption</code>	Server-side encryption algorithm, supports AES256, cos/kms. This field is required when using SSE-COS or SSE-KMS.
None	<code>--sse-customer-algo</code>	Server-side encryption algorithm, supports AES256. This field is required when using SSE-C.
None	<code>--sse-customer-key</code>	Base64-encoded server-side encryption key. For example, <code>MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODIBQkNERUY=</code> . This field is required when using SSE-C.
None	<code>--sse-customer-key-md5</code>	Base64-encoded MD5 hash value of the server-side encryption key. For example, <code>U5L61r7jcwdNvT7frmUG8g==</code> . This field is required when using SSE-C.

None	<code>--checkpoint</code>	Whether to enable checkpoint restart, default is true
------	---------------------------	---

### ⓘ Note:

- When using the command, please ensure that the tool has permission to access the local path.
- `cp` automatically uses concurrent upload/download for large objects.
- If an object is larger than `--part-size`, COSCLI will split the object into multiple parts according to `--part-size` and use `--thread-num` threads to concurrently upload/download the object.
- Each thread maintains a URL. For each URL, you can use the `--rate-limiting` parameter to limit the speed of a single URL. When concurrent upload/download is enabled, the total rate is `--thread-num * --rate-limiting`.
- If an object is upload/download in parts, checkpoint restart will be enabled by default.
- `--include` and `--exclude` support standard regular expression syntax, so you can use them to filter out objects that meet specific criteria.
- By default, symbolic links will not be uploaded. To upload symbolic links, set `--disable-all-symlink=false --enable-symlink-dir=true`. **Note: If there are circular references in the symbolic links, it may cause the tool to enter an infinite loop.**
- When using `zsh`, you may need to add double quotes at both ends of the `pattern` string.

```
./coscli cp ~/test/ cos://bucket1/example/ -r --include
"*.txt$" --meta=x-cos-meta-a:a#ContentType:text#Expires:2022-
10-12T00:00:00.000Z
```

- When using commands in Windows CMD, note that the "—" character (Chinese dash) pasted into CMD will automatically change to "--", and you need to manually enter it.
- For other common options of this command (such as switching bucket and user account), see [Common Options](#).

## Examples

### Upload

#### Uploading a single object

```
./coscli cp ~/example.txt cos://bucket1/example.txt
```

Uploading all files and all folders in the local `test` Directory to the `example` Directory in the `bucket1` bucket

```
./coscli cp ~/test/ cos://bucket1/example/ -r
```

Uploading all `.mp4` files under the local `test` folder and its subfolders to the `example` folder in the `bucket1` bucket

```
./coscli cp ~/test/ cos://bucket1/example/ -r --include ".*\.mp4$"
```

Uploading all non-`.md` files in the local `test` folder and its subfolders to the `example` folder in `bucket1`

```
./coscli cp ~/test/ cos://bucket1/example/ -r --exclude ".*\.md$"
```

Uploading all non-`.md` and non-`.html` files in the local `test` folder and its subfolders to the `example` folder in the `bucket1` bucket

```
./coscli cp ~/test/ cos://bucket1/example/ -r --exclude  
".*\.html$|.*\.md$"
```

Uploading all objects in the `dir` directory (containing the `dirA`, `dirB`, `dirC`, and `dirD` subdirectories) except the `dirD` directory

```
./coscli cp dir/ cos://bucket1/example/ -r --exclude "dirD.*"
```

Uploading all files and folders under the local `test` directory to the `example` directory in the `bucket1` bucket, and store them as archive type files

```
./coscli cp ~/test/ cos://bucket1/example/ -r --storage-class ARCHIVE
```

Uploading the local `file.txt` file to the `bucket1` bucket and setting the single-URL speed limit to 1.3 MB/s

```
./coscli cp ~/file.txt cos://bucket1/file.txt --rate-limiting 1.3
```

## Download

### Downloading a single object

```
./coscli cp cos://bucket1/example.txt ~/example.txt
```

### Downloading all files and folders under the `example` directory in the `bucket1` bucket to the local `test` directory

```
./coscli cp cos://bucket1/example/ ~/test/ -r
```

### Downloading all `.mp4` files in the `example` folder and its subfolders in `bucket1` to the `test` folder on the local device

```
./coscli cp cos://bucket1/example/ ~/test/ -r --include ".*\.mp4$"
```

### Downloading all non-`.md` files under the `example` directory and its subdirectories in `bucket1` to the local `test` directory

```
./coscli cp cos://bucket1/example/ ~/test/ -r --exclude ".*\.md$"
```

### Downloading all non-`.md` and non-`.html` files under the `example` folder and its subfolders in the `bucket1` bucket to the local `test` folder

```
./coscli cp cos://bucket1/example/ ~/test/ -r --exclude  
".*\.html$|.*\.md$"
```

### Download xxx Version of `example.txt` File in `bucket1` to `test` Directory under Local Device

```
./coscli cp cos://bucket1/example.txt ~/test/ --version-id xxx
```

## Copy

### Copying a single object within a bucket

```
./coscli cp cos://bucket1/example.txt cos://bucket1/example_copy.txt
```

## Copying a single object across buckets

```
./coscli cp cos://bucket1/example.txt cos://bucket2/example_copy.txt
```

## Copying all files and folders under the `example1` folder in `bucket1` to the `example2` folder in `bucket2`

```
./coscli cp cos://bucket1/example1/ cos://bucket2/example2/ -r
```

## Copying all `.mp4` data type files under the `example1` folder and its subfolders in the `bucket1` bucket to the `example2` folder in the `bucket2` bucket

```
./coscli cp cos://bucket1/example1/ cos://bucket2/example2/ -r --include  
"*.mp4"
```

## Copying all non-`.md` data type files under the `example1` folder and its subfolders in the `bucket1` bucket to the `example2` folder in the `bucket2` bucket

```
./coscli cp cos://bucket1/example1/ cos://bucket2/example2/ -r --exclude  
"*.md"
```

## Copy xxx Version of `example.txt` file in `bucket1` to `bucket2`

```
./coscli cp cos://bucket1/example.txt cos://bucket2/ --version-id xxx
```

## Move test directory in `bucket1` to `bucket2`

```
./coscli cp cos://bucket1/test/ cos://bucket2/test/ --move -r
```

# Syncing Upload/Download/Copy – sync

Last updated: 2026-03-11 15:27:58

The `sync` command is used to sync object upload, download, and copy. The difference between `sync` and `cp` is that `sync` first compares the CRC64 value of an object with the same name that already exists, and if the value is the same, the object will not be transferred.

## Note:

- If you need to use the upload file command, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucket`, `cos:HeadObject`, `cos:InitiateMultipartUpload`, `cos:UploadPart`, `cos:CompleteMultipartUpload`, `cos:ListMultipartUploads`, `cos:ListParts`.
- If you need to use the download file command, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucket`, `cos:HeadObject`, `cos:GetObject`.
- If you need to use the copy file command, when you perform [authorization policy](#), set the target object action to `cos:GetBucket`, `cos:HeadObject`, `cos:InitiateMultipartUpload`, `cos:PutObject`, `cos:CompleteMultipartUpload`. Set the source object action to `cos:HeadBucket`, `cos:GetBucket`, `cos:HeadObject`, `cos:GetObject`.

For more authorizations, please refer to [CAM-Enabled API](#).

## Command Syntax

```
./coscli sync <source_path> <destination_path> [flag]
```

The `sync` command includes the following parameters:

Parameter Format	Description	Sample
source_path	Source file path, which can be a local path or a COS file path. The COS path is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the <code>endpoint</code> flag.	<ul style="list-style-type: none"> <li>Local path: <code>~/example.txt</code></li> <li>COS file path specified with the bucket alias: <code>cos://bucketalias/example.txt</code></li> <li>COS file path specified with the bucket name: <code>cos://examplebucket-1250000000/example.txt</code></li> </ul>
destination_path	Destination file path, which can be a local path or a COS file path. The COS path is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the <code>endpoint</code> flag.	<ul style="list-style-type: none"> <li>Local path: <code>~/example.txt</code></li> <li>COS file path specified with the bucket alias: <code>cos://bucketalias/example.txt</code></li> <li>COS file path specified with the bucket name: <code>cos://examplebucket-1250000000/example.txt</code></li> </ul>

`sync` includes the following optional flags:

Flag Abbreviation	Flag Name	Description
None	<code>--include</code>	<ul style="list-style-type: none"> <li>Includes specific objects (Versions prior to v1.0.4 only filter the local file name during upload, while versions v1.0.4 and later will filter the full path.)</li> <li>For example: upload all files under <code>./test</code> to COS. The <code>./test</code> folder contains the <code>aaa</code> folder, which includes the <code>1.txt</code> file. <ul style="list-style-type: none"> <li>○ Versions prior to 1.0.4 will match <code>aaa/1.txt</code></li> <li>○ 1.0.4 and later versions will match <code>./test/aaa/1.txt</code></li> </ul> </li> </ul>
None	<code>--exclude</code>	<ul style="list-style-type: none"> <li>Excludes specific objects. (Versions prior to v1.0.4 only filter the local file name during upload, while versions v1.0.4 and later will filter the full path.)</li> <li>For example: upload all files under <code>./test</code> to COS. The <code>./test</code> folder contains the <code>aaa</code> folder, which includes the <code>1.txt</code> file. <ul style="list-style-type: none"> <li>○ Versions prior to 1.0.4 will match <code>aaa/1.txt</code></li> <li>○ 1.0.4 and later versions will match <code>./test/aaa/1.txt</code></li> </ul> </li> </ul>
<code>-r</code>	<code>--recursive</code>	Specifies whether to traverse all objects in the directory recursively.
None	<code>--storage-class</code>	Specifies the storage class of the uploaded file. Default value: <code>STANDARD</code> . For more information, see <a href="#">Storage Class Overview</a> .
None	<code>--part-size</code>	The chunk size of the file, measured in MB, has a default value of 32 and a maximum support of 5120. To adapt the chunk size based on the file size, please set it to 0.
None	<code>--thread-num</code>	Number of concurrent threads. Default value: 5
None	<code>--rate-limiting</code>	Speed limit for a single URL. Value range: 0.1-100 MB/s
None	<code>--snapshot-path</code>	Specifies the directory where the snapshot information is stored when the uploaded or downloaded file is saved. During the next file upload or download, COSCLI will read the snapshot information in the specified directory for incremental upload or download. This option is used to speed up directory file sync.
		Metadata of the uploaded file, including certain HTTP standard

None	<code>--meta</code>	metadata of the uploaded file, including certain HTTP standard attributes (HTTP Header) and custom metadata prefixed with <code>x-cos-meta-</code> (User Meta). The file metadata is in the format of <code>header:value#header:value</code> , such as <code>Expires:2022-10-12T00:00:00.000Z#Cache-Control:no-cache#Content-Encoding:gzip#x-cos-meta-x:x</code> .
None	<code>--routines</code>	Specifies the number of files for concurrent upload or download threads between files. The default value is <code>3</code> .
None	<code>--fail-output</code>	This option determines whether to enable the error output of files when upload or download fails (The default value is <code>true</code> , enabled). If enabled, failed file transfers will be recorded in the specified directory (If no directory is specified, the default directory <code>./coscli_output</code> is used). If disabled, only the number of failed files will be output to the console.
None	<code>--fail-output-path</code>	This option is used to specify the error output folder for recording failed file uploads or downloads. By providing a custom folder path, you can control the location and name of the error output folder. If this option is not set, the default error log folder <code>./coscli_output</code> is used.
None	<code>--retry-num</code>	Number of frequency limit retry times (The default value is <code>0</code> , indicating no retry will be performed). It can be specified as <code>1-10</code> times. When multiple machines execute download operations on the same COS directory simultaneously, you can specify this parameter to retry to avoid frequency limit errors.
None	<code>--err-retry-num</code>	Number of error retry times (The default value is <code>5</code> ). Set it as <code>1-10</code> times, or set it to <code>0</code> , indicating no retry will be performed.
None	<code>--err-retry-interval</code>	Retry interval (only available when <code>--err-retry-num</code> is set as <code>1-10</code> ). Set the retry interval as <code>1-10</code> seconds. If it is not set or is set to <code>0</code> , each retry interval will be a random value between <code>1-10</code> seconds.
None	<code>--only-current-dir</code>	Whether to upload only files in the current directory, ignoring subdirectories and their contents (The default value is <code>false</code> , indicating they will not be ignored).
None	<code>--disable-</code>	Whether to ignore the subfiles and subdirectories of all the soft links during upload (The default value is <code>true</code> , indicating they

e	all-symlink	will not be ignored). Currently only supported on Linux and MacOS systems
None	--enable-symlink-dir	Whether to upload the subdirectories of the soft links (The default value is <code>false</code> , indicating they will not be uploaded). Currently only supported on Linux and MacOS systems
None	--disable-crc64	Whether to disable CRC64 data validation (The default value is <code>false</code> , indicating validation is enabled).
None	--delete	Deletes any other files in the specified target path, retaining only the files synchronized this time (The default value is <code>false</code> , indicating the files will not be deleted). It is recommended to enable version control before using the <code>--delete</code> option to prevent accidental data deletion.
None	--backup-dir	Synchronizes the backup of deleted files, which preserves files that are deleted on the target end but do not exist on the source end (effective only during the download, and must be specified when <code>--delete=true</code> ). For upload and bucket copies, use version control to restore accidentally deleted data.
None	--force	Forced operation, without confirmation prompt (The default value is <code>false</code> ).
None	--disable-checksum	The default is true, verifying only the fragment crc64. When set to false, it verifies the entire file crc64. (coscli V1.0.6 and prior versions default to false)
None	--process-log	Whether process log is enabled, the default is true, enabled
None	--process-log-path	This option is used to specify a dedicated output folder to store process logs. The logs will record information related to file upload or download, including error logs, normal execution logs, retry details. By providing a custom folder path, you can control the location and name of the log output folder. If not set, the default log folder ( <code>coscli_output</code> ) will be used.
None	--update	Transmit only files with a newer last modified time
None	--ignore-existing	Transmit only non-existent file
None	--skip-dir	Defaults to false. When set to true, it skips folders during transmission

e		transmission.
None	--acl	Set the file's ACL, such as private, public-read.
None	--grant-read	Grant the authorized entity permission to read the object. The format is id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-read-acp	Grant the authorized entity permission to read the object's access control list (ACL). The format is id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-write-acp	Grant the authorized entity permission to write to the object's access control list (ACL). The format is id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-full-control	Grant the authorized entity all privileges on the operation object. The format is id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--tags	The object tag collection supports up to 10 tags (for example, --tags="Key1=Value1&Key2=Value2")
None	--forbid-overwrite	<p>For buckets with Versioning Not Enabled, when uploading files, specify whether to forbid overwriting objects with the same name.</p> <ul style="list-style-type: none"> <li>• When set to false, default overwrite for objects with the same name.</li> <li>• When set to true, it means overwriting an Object with the same name is forbidden.</li> </ul> <p>When the bucket is in enabled or paused versioning status, the x-cos-forbid-overwrite header setting is invalid, allowing overwriting objects with the same name.</p>
None	--encryption-type	Server-side encryption method (SSE-COS/SSE-C).
None	--server-side-	Server-side encryption algorithm, supports AES256, COS/KMS

e	server-side encryption	Server-side encryption. This field is required when using SSE-COS or SSE-KMS.
None	<code>--sse-customer-algo</code>	Server-side encryption algorithm, supports AES256. This field is required when using SSE-C.
None	<code>--sse-customer-key</code>	Base64-encoded server-side encryption key. For example, <code>MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODIBQkNERUY=</code> . This field is required when using SSE-C.
None	<code>--sse-customer-key-md5</code>	Base64-encoded MD5 hash value of the server-side encryption key. For example, <code>U5L61r7jcwdNvT7frmUG8g==</code> . This field is required when using SSE-C.
None	<code>--checkpoint</code>	Whether to enable checkpoint restart, default is true

**Note:**

- Please ensure the tool has permission to access local paths when commands are used.
- `sync` automatically uses concurrent upload/download for large objects.
- If an object is larger than `--part-size`, COSCLI will split the object into multiple parts according to `--part-size` and use `--thread-num` threads to concurrently upload/download the object.
- Each thread maintains a URL. For each URL, you can use the `--rate-limiting` parameter to limit the speed of a single URL. When concurrent upload/download is enabled, the total rate is `--thread-num * --rate-limiting`.
- If an object is uploaded/downloaded in parts, checkpoint restart will be enabled by default.
- `--include` and `--exclude` support standard regular expression syntax, so you can use them to filter out objects that meet specific criteria.
- By default, symlinks are not uploaded. To upload symlinks, set `--disable-all-symlink=false --enable-symlink-dir=true`. **Note: If a symlink has a circular reference, the tool may enter an infinite loop.**
- When using `zsh`, you may need to add double quotes at both ends of the `pattern` string.
- Do not set `snapshot-path` to the directory to be migrated or its subdirectories.

```
./coscli sync ~/test/ cos://bucket1/example/ -r --include  
".*\.txt$" --snapshot-path=/path/snapshot-path --meta=x-cos-  
meta-a:a#ContentType:text#Expires:2022-10-12T00:00:00.000Z
```

- When you use the sync command, in addition to PUT request fees, there are two other scenarios where you will have a HEAD request for cloud files, which will incur additional fees:
  - If the parameter to specify the snapshot directory ( `--snapshot-path` ) is not added, HEAD request fees will be incurred.
  - If the parameter to specify the snapshot directory ( `--snapshot-path` ) is added and it is the first time the snapshot directory is generated, HEAD request fees will be incurred. If it is not the first time the snapshot directory is generated, no additional request fees will be incurred.
- For other common options of this command (such as switching bucket and user account), see [Common Options](#).

## Sample

### Syncing object upload

```
./coscli sync ~/example.txt cos://bucket1/example.txt
```

### Syncing object download

```
./coscli sync cos://bucket1/example.txt ~/example.txt
```

### Syncing intra-bucket replication

```
./coscli sync cos://bucket1/example.txt cos://bucket1/example_copy.txt
```

### Syncing cross-bucket replication

```
./coscli sync cos://bucket1/example.txt cos://bucket2/example_copy.txt
```

# Deleting Objects – rm

Last updated: 2025-05-12 15:10:50

The `rm` command is used to delete an object.

## ⚠ Note:

- To use the `rm` command, please download COSCLI V1.0.1 or above. For details, see [Download and Installation Configuration](#).
- If your current version of COSCLI is V1.0.0, please upgrade to V1.0.1 before executing the `rm` command. In V1.0.0, the `--include` and `--exclude` parameters do not work when executing the `rm` command, which may lead to unexpected deletion situations.
- If you need to use this command to delete an object, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:HeadObject`, `cos:GetBucket`, `cos>DeleteObject`, `cos>DeleteMultipleObjects`.
- If you need to use this command to delete a historical version (such as importing `--all-versions` or `--version-id`), when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:HeadObject`, `cos:GetBucket`, `cos>DeleteObject`, `cos>DeleteMultipleObjects`, `cos:GetBucketVersioning`, `cos:GetBucketObjectVersions`.

For more authorizations, please refer to [CAM-Enabled API](#). If you have any questions or need further assistance, [please contact us](#).

## Command format

```
./coscli rm cos://<bucket-name>[/prefix/] [flag]
```

`rm` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;bucket-name&gt;</code>	Specify the bucket to access. You can use the bucket alias from the <a href="#">configuration parameters</a> or the bucket name. If using the bucket name, you must also include the <code>endpo</code> <code>int</code> flag.	Access using bucket alias: <code>cos://example-alias</code> Access using bucket name: <code>cos://examplebucket-1250000000</code>
<code>/prefix/</code>	Specifies a directory (optional).	<code>/picture/</code>

`rm` includes the following optional flags:

Flag Abbreviation	Flag Name	Description
<code>-h</code>	<code>--help</code>	Views the usage of this command.
None	<code>--include</code>	Includes specific objects.
None	<code>--exclude</code>	Excludes specific objects.
<code>-r</code>	<code>--recursive</code>	Specifies whether to traverse all objects in the directory recursive.
<code>-f</code>	<code>--force</code>	Force deletion (no prompt before the deletion).
None	<code>--fail-output</code>	This option determines whether to enable the error output of files when upload or download fails (The default value is <code>true</code> , enabled). If enabled, failed file transfers will be recorded in the specified directory (If no directory is specified, the default directory <code>./coscli_output</code> is used). If disabled, only the number of failed files will be output to the console.
None	<code>--fail-output-path</code>	This option is used to specify the error output folder for recording failed file uploads or downloads. By providing a custom folder path, you can control the location and name of the error output folder. If this option is not set, the default error log folder <code>./coscli_output</code> is used.
None	<code>--all-versions</code>	Available only in buckets with versioning enabled and when importing the <code>--recursive (-r)</code> parameter, traverse and delete all versions under the specified path.
None	<code>--version-id</code>	Available only in a Bucket with versioning enabled and when <code>--recursive (-r)</code> parameter is not passed in, to delete the specified version of a specified object.

**Note:**

- `--include` and `--exclude` support standard regular expressions. You can use regular expressions to filter objects that meet your requirements.

- When using `zsh`, you may need to enclose the pattern string with double quotation marks.

```
./coscli rm cos://bucket1/example/ -r --include ".*\.mp4$"
```

- For more general options for this command (such as switching buckets or user accounts), see [General Options](#).

## Sample

### Deleting the `fig1.png` object

```
./coscli rm cos://bucket1/fig1.png
```

### Deleting all objects in the `picture` directory

```
./coscli rm cos://bucket1/picture/ -r
```

### Deleting the Specified Version of the `fig1.png` File

```
./coscli rm cos://bucket1/fig1.png --version-id xxx
```

### Deleting All Versions with the `test` Prefix

```
./coscli rm cos://bucket1/test -r --all-versions
```

# Getting File Hash Value – hash

Last updated: 2025-05-12 15:11:10

The `hash` command is used to calculate the hash value of a local file or get the hash value of a file in COS.

## Note:

If you need to use this command, when you perform [authorization policy](#), set action to `cos:HeadObject`. For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli hash <object-name> [flag]
```

`hash` includes the following parameters:

Parameter Format	Description	Sample
<code>&lt;object-name&gt;</code>	Specifies the file to access. It can be a local path or a COS file path. COS paths support using the bucket alias or bucket name from the <a href="#">Configuration Parameters</a> . If accessing with the bucket name, you need to carry the <code>enpoint</code> flag additionally.	Local path: <code>~/example.tx</code> Specify the COS file path using the bucket alias: <code>cos://bucketalias/example.txt</code> Specify the COS file path using the bucket name: <code>cos://examplebucket-1250000000/example.txt</code>

The `hash` command contains the following optional flags:

Flag Abbreviation	Flag Name	Description
<code>-h</code>	<code>--help</code>	Views the usage of this command.
<code>-</code>	<code>--type</code>	Hash type, which can be MD5 or CRC64 (default) Note: The md5 can only obtain the etag. If you need to get the file md5, you can download the complete file and then calculate it.

**Note:**

For other general options related to this command (such as switching buckets or user accounts), see the [General Options](#) documentation.

## Sample

### Calculating the CRC64 value of local file

```
./coscli hash ~/test.txt
```

### Getting the MD5 value of COS file

```
./coscli hash cos://bucket1/example.txt --type=md5
```

# Listing Incomplete Multipart Uploads

## – lsparts

Last updated: 2025-05-12 15:11:26

The `lsparts` command is used to list the generated incomplete multipart uploads.

### Note:

- If you need to use this command to list multipart upload tasks, when you perform [authorization policy](#), set action to `cos:PutBucket`.
- If you need to use this command to list the fragments of the corresponding file of the multipart upload task, when you perform [authorization policy](#), set action to `cos:ListMultipartUploads`, `cos:ListParts`.

For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli lsparts cos://<bucket-name>[/prefix/] [flag]
```

`lsparts` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;bucket-name&gt;</code>	Specify the bucket to access. You can use the bucket alias from the <a href="#">configuration parameters</a> or the bucket name. If using the bucket name, you must also include the <code>end point</code> flag.	Access using bucket alias: <code>cos://example-alias</code> Access using bucket name: <code>cos://examplebucket-12500000000</code>
<code>/prefix/</code>	Specifies a directory (optional).	<code>/picture/</code>

The `lsparts` command contains the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--include	Includes specific objects.
None	--exclude	Excludes specific objects.
None	--limit	Specifies the maximum quantity (0-1000) to be listed
None	--upload-id	Upload record ID. When uploading the uploadid, you need to specify the specific file. Only show the fragment information of the file corresponding to the uploadid. If the uploadid is not passed in, show the parts being uploaded under the specified prefix.

**Note:**

- For more information on multipart uploads, see [Multipart Uploads](#).
- For other general options related to this command (such as switching buckets or user accounts), see the [General Options](#) documentation.

## Sample

### Listing all incomplete multipart uploads in `bucket1` bucket

```
./coscli lsparts cos://bucket1
```

The returned information includes the object key (the unique identifier of the object in the bucket), multipart upload ID, multipart upload start time, and total number of incomplete multipart uploads in the bucket. Below is an example:

```

      KEY           |                UPLOAD ID                |          INITIATE
      TIME
-----+-----+-----
      test.txt     | 1671191183635d2b71b1d68a0** | 2022-01-01T00:00:00.000Z
-----+-----+-----

```

TOTAL: 1

## Listing All Uploaded Fragments of the test.txt File in bucket1 with upload\_id 1671191183635d2b71b1d68a0\*\*\*\*\*

```
./coscli lsparts cos://bucket1/test.txt --upload-id="1671191183635d2b71b1d68a0*****"
```

The returned result is as follows. The output information includes: PARTNUMBER (chunk number), ETAG (MD-5 algorithm checksum of the chunk), LastModified (last modified time of the chunk), SIZE (size of each part).

```

PARTNUMBER |          ETAG          |          LAST MODIFIED
|  SIZE
-----+-----+-----
1          | "58f06dd588d8ffb3beb46ada6309436b" | 2024-12-
17T16:34:48+08:00 | 32.00 MB
2          | "58f06dd588d8ffb3beb46ada6309436b" | 2024-12-
17T16:34:48+08:00 | 32.00 MB
3          | "58f06dd588d8ffb3beb46ada6309436b" | 2024-12-
17T16:34:48+08:00 | 32.00 MB
4          | "58f06dd588d8ffb3beb46ada6309436b" | 2024-12-
17T16:34:48+08:00 | 32.00 MB
-----+-----+-----

TOTAL: 4
-----

```

# Clearing Incomplete Multipart Uploads – abort

Last updated: 2025-05-12 15:11:41

The `abort` command is used to clear the generated incomplete multipart uploads.

## Note:

If you need to use this command, when you perform [authorization policy](#), set action to `cos:ListMultipartUploads`, `cos:AbortMultipartUpload`. For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli abort cos://<bucket-name>[/prefix/] [flag]
```

`abort` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;bucket-name&gt;/&lt;key&gt;</code>	Specify the object in the bucket. Support using the <a href="#">configuration parameter</a> bucket alias or bucket name for access. If accessing with the bucket name, you must also carry the <code>endpoint</code> flag.	Access using bucket alias: <code>cos://example-alias</code> Access using bucket name: <code>cos://examplebucket-1250000000</code>
<code>/prefix/</code>	Specifies a directory (optional).	<code>/picture/</code>

The `abort` command contains the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--include	Includes specific objects.
None	--exclude	Excludes specific objects.
None	--fail-output	This option determines whether to enable error output when cleaning up file fragments (enabled by default <code>true</code> ). If enabled, file fragment cleanup failures will be recorded in the specified directory (if not specified, defaults to <code>./coscli_output</code> ). If disabled, only the number of cleanup failures will be output to the console.
None	--fail-output-path	This option is used to specify the error output folder for recording during file fragment cleanup. By providing a custom folder path, you can control the location and name of the error output folder. If this option is not set, the default error log folder <code>./coscli_output</code> will be used.

### Note

For other general options related to this command (such as switching buckets or user accounts), see the [General Options](#) documentation.

## Sample

### Clearing all incomplete multipart uploads in `bucket1` bucket

```
./coscli abort cos://bucket1
```

### Clear all fragment within the `picture` folder in the `bucket1` bucket.

```
./coscli abort cos://bucket1/picture/
```

# Retrieving Archived Files – restore

Last updated: 2025-05-12 15:13:47

The `restore` command is used to retrieve archive files.

## Note:

If you need to use this command, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucket`, `cos:PostObjectRestore`. For more authorizations, please refer to [CAM-Enabled API](#).

## Command format

```
./coscli restore cos://<bucket-name>[/prefix/] [flag]
```

`restore` includes the following parameters:

Parameter Format	Description	Sample
<code>cos://&lt;bucket-name&gt;</code>	Specify the bucket to access. You can use the bucket alias from the <a href="#">configuration parameters</a> or the bucket name. If using the bucket name, you must also include the <code>endpoint</code> flag.	Access using bucket alias: <code>cos://example-alias</code> Access using bucket name: <code>cos://examplebucket-1250</code> <code>000000</code>
<code>/prefix/</code>	Specifies a directory (optional).	<code>/picture/</code>

The `restore` command contains the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--include	Includes specific objects.
None	--exclude	Excludes specific objects.
-d	--days	Specifies the expiration time of the temporary file generated during object restoration, which is 3 days by default.
-m	--mode	Specifies the restoration mode, which is <code>Standard</code> by default. <ul style="list-style-type: none"> <li>If the data to be recovered is of the archive storage type, the available values are Expedited, Standard, and Bulk, which correspond to expedited retrieval mode, standard retrieval mode, and bulk retrieval mode respectively.</li> <li>If the data being restored is of deep archive storage type, the optional values are Standard, Bulk.</li> </ul>
-r	--recursive	Traverse the folder recursive.
None	--fail-output	This option determines whether to enable the error output of files when reheating fails (enabled by default <code>true</code> ). If enabled, the failure of file reheating will be recorded in the specified directory (if not specified, defaults to <code>./coscli_output</code> ). If disabled, only the quantity of error files will be output to the console.
None	--fail-output-path	This option is used to specify the error output folder for recording failed reheating files. By providing a custom folder path, you can control the location and name of the error output folder. If this option is not set, the default error log folder <code>./coscli_output</code> will be used.

### ⓘ Note

- `--include` and `--exclude` support standard regular expressions. You can use regular expressions to filter objects that meet your requirements.

- When using `zsh`, you may need to enclose the pattern string with double quotation marks.

```
./coscli restore cos://bucket1/example/ -r --include ".*\.mp4$"
```

- For more general options for this command (such as switching buckets or user accounts), see [General Options](#).
- For more information on retrieving archive files, please refer to [POST Object restore – API Documentation](#).

## Sample

### Restoring archive objects in `bucket1` bucket in Standard mode

```
./coscli restore cos://bucket1/picture.jpg
```

### Retrieve all archive files in the `picture` folder within the `bucket1` bucket using expedited retrieval mode

```
./coscli restore cos://bucket1/picture/ -r --mode Expedited
```

# Getting Pre-signed URL – signurl

Last updated: 2026-03-11 15:26:40

The `signurl` command is used to get the pre-signed URL of an object, through which the object can be accessed anonymously.

## Command Syntax

```
./coscli signurl cos://<bucket-name>/<key> [flag]
```

`signurl` includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;/&lt;key&gt;</code>	Specifies the target object in the bucket, which is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias/test.txt</code> Access with the bucket name: <code>cos://examplebucket-1250000000/test.txt</code>

The `signurl` command contains the following optional flags:

Flag Abbreviation	Flag Name	Description
<code>-h</code>	<code>--help</code>	Views the usage of this command.
<code>-t</code>	<code>--time</code>	Sets the URL expiration time, which is 1000s by default
None	<code>--simple-output</code>	Enables the simple output mode, with a default value of false. When set to true, only the signature URL is output to simplify the returned result.

### Note:

For other common options of this command (such as switching bucket and user account), see [Common Options](#).

## Examples

### Getting the pre-signed URL of `picture.jpg` in `bucket1` bucket

```
./coscli signurl cos://bucket1/picture.jpg
```

The returned results are as follows:

```
INFO[2026-02-04 11:36:10] Signed URL:
INFO[2026-02-04 11:36:10] https://bucket1-1234124.cos.ap-
guangzhou.myqcloud.com/picture.jpg?q-sign-algorithm=sha1&q-
ak=*****&q-sign-
time=1770176170%3B1770186170&q-key-time=1770176170%3B1770186170&q-
header-list=host&q-url-param-list=&q-
signature=*****
```

### Getting the pre-signed URL of `picture.jpg` in `bucket2` bucket and setting the URL expiration time to 1314s

```
./coscli signurl cos://bucket2/picture.jpg --time 1314
```

The returned results are as follows:

```
INFO[2026-02-04 11:36:10] Signed URL:
INFO[2026-02-04 11:36:10] https://bucket2-1234124.cos.ap-
guangzhou.myqcloud.com/picture.jpg?q-sign-algorithm=sha1&q-
ak=*****&q-sign-
time=1770176170%3B1770186170&q-key-time=1770176170%3B1770186170&q-
header-list=host&q-url-param-list=&q-
signature=*****
```

### Generate a pre-signed URL for `picture.jpg` in `bucket3` and Output in Simple Mode

```
./coscli signurl cos://bucket3/picture.jpg --simple-output
```

The returned results are as follows:

```
https://bucket3-1234124.cos.ap-guangzhou.myqcloud.com/picture.jpg?q-  
sign-algorithm=sha1&q-ak=*****&q-sign-  
time=1770176267%3B1770186267&q-key-time=1770176267%3B1770186267&q-  
header-list=host&q-url-param-list=&q-  
signature=*****
```

# Creating/Obtaining a Symbolic Link – symlink

Last updated: 2025-05-12 15:13:10

The `symlink` command is used to create or obtain a symbolic link to an object. You can quickly find the object through this symbolic link.

## Note:

- If you need to use the command to create a symbolic link, when you perform [authorization policy](#), set action to `cos:PutBucket`.
- If you need to use the command to obtain a symbolic link, when you perform [authorization policy](#), set action to `cos:GetObject`.

For more authorizations, please refer to [CAM-Enabled API](#).

## Command Syntax

```
./coscli symlink --method create cos://<bucket-name>/<key> --link
linkKey [flag]
```

The `symlink` command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;</code>	Specifies the target bucket, which is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias</code> Access with the bucket name: <code>cos://examplebucket-1250000000</code>
<code>/&lt;key&gt;</code>	Specifies the object name.	<code>/test</code>

The `symlink` command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
No	--method	Valid values: create/get.
No	--link	Specifies the symbolic link key value.

## Operation Example

### Creating a symbolic link named symlink for picture.jpg in the bucket1 bucket

```
./coscli symlink --method create cos://bucket1/picture.jpg --link  
symlink
```

### Accessing the target object via the symlink

```
./coscli symlink --method get cos://bucket2 --link symlink
```

# Viewing Contents of an Object – cat

Last updated: 2025-05-12 15:12:53

The cat command is used to view the contents of an object.

## Note:

If you need to use this command, when you perform [authorization policy](#), set action to `cos:GetBucket`. For more authorizations, please refer to [CAM-Enabled API](#).

## Command Syntax

```
./coscli cat cos://<bucket-name>/<key> [flag]
```

The cat command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;/&lt;key&gt;</code>	Specifies the target bucket, which is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the endpoint flag.	Access with the bucket alias: <code>cos://example-alias/test.txt</code> Access with the bucket name: <code>cos://examplebucket-1250000000/test.txt</code>

The cat command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
<code>-h</code>	<code>--help</code>	Views the usage of this command.

## Operation Example

### Viewing the contents of the test.txt file in the bucket1 bucket

```
./coscli cat cos://bucket1/test.txt
```

# Listing Contents and Statistics Under a Directory – lsdu

Last updated: 2025-05-12 15:12:37

The lsdu command is used to obtain the contents of the current level with a specified prefix and to calculate the size and number of objects within it.

## Note:

If you need to use this command, when you perform [authorization policy](#), set action to `cos:HeadBucket`, `cos:GetBucket`. For more authorizations, please refer to [CAM-Enabled API](#).

## Command Syntax

```
./coscli lsdu cos://<bucket-name>[/prefix/] [flags]
```

The lsdu command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;</code>	Specifies the target bucket, which is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias</code> Access with the bucket name: <code>cos://examplebucket-1250000000</code>
<code>/prefix/</code>	Specifies a directory (optional).	<code>/picture/</code>

The lsdu command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--include	Includes specific objects.
None	--exclude	Excludes specific objects.

## Operation Example

### Obtaining the statistics under the root directory of the bucket1 bucket

```
./coscli lsdu cos://bucket1/
```

The returned information includes the names of directories or objects, the total number of objects, and the total size. Below is an example:

```

      NAME           | OBJECTS COUNT | TOTAL SIZE
-----+-----+-----
      300123/       |           1   | 300.00 MB
      300s/         |           1   | 100.00 MB
      300u/         |          301  |   8.22 GB
      activity/    |           35  | 129.08 KB
      test/         |            1  |         3 B
      test100/     |           20  | 20.00 GB
      test5/       |            6  |   9.00 GB
      testrm/      |            1  |         0 B
      10GB_file    |            1  | 11.72 GB
      1GB_file     |            1  |   1.00 GB

```

### Obtaining the statistics under the picture directory of the bucket1 bucket

```
./coscli lsdu cos://bucket1/picture/
```

#### Note:

When there are many files, this command may take a long time to run. It is recommended to run it in the background.

# Bucket Versioning – bucket-versioning

Last updated: 2025-05-12 15:12:18

The bucket-versioning command is used to manage versioning for a bucket. It can be used to enable, suspend, and view the status of bucket versioning.

## Note:

- If you need to use the command to enable/suspend versioning, when you perform [authorization policy](#), set action to `cos:PutBucketVersioning`.
- If you need to use the command to view the bucket versioning status, when you perform [authorization policy](#), set action to `cos:GetBucketVersioning`.

For more authorizations, please see [CAM-Enabled API](#).

## Command Syntax

The following bucket-versioning commands are used to enable and suspend versioning:

```
./coscli bucket-versioning --method [method] cos://<bucket-name>
versioning
```

The bucket-versioning and its subcommands include the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
-c	--method	Valid values: put/get.

## Operation Example

### Enable Bucket Versioning

```
./coscli bucket-versioning --method put cos://examplebucket Enabled
```

### Suspend Bucket Versioning

```
./coscli bucket-versioning --method put cos://examplebucket Suspended
```

## View Bucket Versioning Status

```
./coscli bucket-versioning --method get cos://examplebucket
```

# Object Tag – object-tagging

Last updated: 2025-10-28 11:29:36

The object-tagging command is used to create (modify), query, and delete object tags. Each object supports up to 10 group tags.

## Note:

- To query object tags, when you perform [authorization policy](#), set the action to `cos:GetObjectTagging`.
- To set object tags, when performing [authorization policy](#), set the action to `cos:PutObjectTagging`.
- To delete object tags, when performing [authorization policy](#), set the action to `cos:DeleteObjectTagging`.

For more authorization, please refer to [business APIs that support CAM](#).

## Command Syntax

```
./coscli object-tagging --method [method] cos://<bucket-name>/object  
[tag_key]#[tag_value]
```

The object-tagging command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;</code>	Specify the target bucket. Supports using the bucket alias or bucket name in <a href="#">configuration parameters</a> . If using the bucket name for access, you must also carry the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias</code> Access with the bucket name: <code>cos://examplebucket-1250000000</code>

The object-tagging command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--method	Specify the operation to perform, including put (set tag), get (query tag), delete (delete tag), and add (add tag).
None	--version-id	Specified version of the operation object

**Note:**

For more general options for this command (such as switching buckets or user accounts), see [Common Options](#).

## Modifying Object Tags

Object tags are represented by a set of Key-Value pairs. Only the object owner and users with PutObjectTagging permissions can add or modify object tags. Otherwise, error code 403 AccessDenied will be returned.

### Command Syntax

```
./coscli object-tagging --method put cos://bucketAlias/object
key1#value1 key2#value2
```

Among them, `key#value` represents the tag key-value pair Key-Value, where key and value are separated by `#`. If the object has no tags, this command will add the designated tags to the object. If the object already has tags, this command will overwrite the original tags.

### Operation Example

Configure two group tags for the object under bucket alias example-alias. Among them, one group tag has key 1 and value 111, the other group tag has key 2 and value 222. The command is as follows:

```
./coscli object-tagging --method put cos://example-alias/object 1#111
2#222
```

## GET Object tagging

### Command Syntax

```
./coscli object-tagging --method get cos://bucketAlias/object
```

### Operation Example

```
./coscli object-tagging --method get cos://example-alias/object
```

The output result shows the object under bucket alias example-alias is configured with two group tags. Among them, one tag has key 1 and value 111, the other has key 2 and value 222.

```
KEY | VALUE  
-----+-----  
  1 |  111  
  2 |  222
```

## Deleting Object Specified Tag

### Command Syntax

```
./coscli object-tagging --method delete cos://bucketAlias/object  
key1#value1
```

### Operation Example

```
./coscli object-tagging --method delete cos://example-alias/object 1#111
```

## Deleting ALL Object Tags

### Command Syntax

```
./coscli object-tagging --method delete cos://bucketAlias/object
```

### Operation Example

```
./coscli object-tagging --method delete cos://example-alias/object
```

## Adding Object Tags

Object tags are represented by a set of Key-Value pairs. Only the object owner and users with PutObjectTagging permissions can add or modify object tags. Otherwise, error code 403 AccessDenied will be returned.

## Command Syntax

```
./coscli object-tagging --method add cos://bucketAlias/object  
key3#value3
```

Among them, `key#value` represents the tag key-value pair Key-Value, where key and value are separated by `#`.

## Operation Example

Add a group tag to the object under bucket alias example-alias. The tag has key 1 and value 111. The command is as follows:

```
./coscli object-tagging --method add cos://example-alias/object 1#111
```

# Management Object ACL – object-acl

Last updated: 2025-10-28 11:32:53

The object-acl command is used to set and query object ACL.

## Note:

- To query object acl, when performing [authorization policy](#), set action to `cos:GetObjectACL`.
- To set object acl, when performing [authorization policy](#), set action to `cos:PutObjectACL`.

For more authorization, please refer to [Business APIs Supporting CAM](#).

## Command Syntax

```
./coscli object-acl --method [method] cos://<bucket-name>/object
```

The object-acl command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;</code>	Specify the target bucket. Supports using the bucket alias in <a href="#">configuration parameters</a> or the bucket name for access. If using the bucket name for access, you must also include the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias</code> Access with the bucket name: <code>cos://examplebucket-1250000000</code>

The object-acl command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--method	Specify the required operation, including put (set object ACL), get (object ACL query).
None	--acl	Set the file's ACL, such as private, public-read.
None	--grant-read	Grant the read permission of the object to the authorized. Format: id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-read-acp	Grant the authorized permission to read the object's access control list (ACL). Format: id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-write-acp	Grant the authorized permission to write to the object's access control list (ACL). Format: id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-full-control	Grant all permissions on the operation object to the authorized. Format: id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".

**Note:**

For more general options for this command (such as switching buckets or user accounts), see [Common Options](#).

## Operation Example

### Setting an ACL for an Object





# Managing Bucket ACL – bucket-acl

Last updated: 2025-10-28 11:33:50

The `bucket-acl` command is used to set or query bucket ACL.

## Note:

- To query bucket ACL, when performing [authorization policy](#), set action to `cos:GetBucketACL`.
- To set bucket ACL, when performing [authorization policy](#), set action to `cos:PutBucketACL`.

For more authorization, please refer to [Business APIs Supporting CAM](#).

## Command Syntax

```
./coscli bucket-acl --method [method] cos://<bucket-name>
```

The `bucket-acl` command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;</code>	Specifies the target bucket, which is accessible by using the bucket alias or bucket name configured in the configuration file as detailed in <a href="#">Download and Installation Configuration</a> . If you use the bucket name for access, you also need to include the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias</code> Access with the bucket name: <code>cos://examplebucket-1250000000</code>

The `bucket-acl` command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--method	Specify the required operation, including put (set bucket ACL), get (query bucket ACL).
None	--acl	Set the file's ACL, such as private, public-read, public-read-write.
None	--grant-read	Grant the authorized entity permission to read the bucket. Format: id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-read-acp	Grant the authorized entity permission to read the bucket's access control list (ACL). Format: id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-write-acp	Grant the authorized entity permission to write to the bucket's access control list (ACL). Format: id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".
None	--grant-full-control	Grant the authorized entity all permissions to operate the bucket. Format: id="[OwnerUin]", for example id="100000000001". Use comma (half-width) to separate multiple authorized entities, such as id="100000000001",id="100000000002".

**Note:**

For more general options for this command (such as switching buckets or user accounts), see [Common Options](#).

## Setting a Bucket ACL

### Operation Example



```

+-----+
+
+ | Display Name |
+-----+-----+-----+
+
+ |           |
+-----+-----+-----+
+
Grant #2 | Permission | READ
+-----+-----+-----+
+
+ | Grantee Type | CanonicalUser
+-----+-----+-----+
+
+ | ID           | qcs::cam::uin/100000000012:uin/100000000012
+-----+-----+-----+
+
+ | Display Name |
+-----+-----+-----+

```

-  
Access Control List (ACL) Information

Summary:

- Owner: qcs::cam::uin/1000000000:uin/1000000000 (UIN: )
- Total Grants: 2
- Permissions:
  - READ: 2 grants

# Bucket Policy – bucket-policy

Last updated: 2025-10-28 11:35:02

The bucket-encryption command is used to set, query, and delete bucket encryption policies.

## Note:

- To query bucket encryption policy, when performing [authorization policy](#), set action to `cos:GetBucketEncryption`.
- To configure bucket encryption policy, when performing [authorization policy](#), set action to `cos:PutBucketEncryption`.
- To delete bucket encryption policy, when performing [authorization policy](#), set action to `cos>DeleteBucketEncryption`.

For more authorization, please refer to [Business APIs Supporting CAM](#).

## Command Syntax

```
./coscli bucket-encryption --method [method] cos://<bucket-name>
```

The bucket-encryption command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;</code>	Specify the target bucket. Supports using the bucket alias in <a href="#">configuration parameters</a> or the bucket name for access. If using the bucket name for access, you must also include the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias</code> Access with the bucket name: <code>cos://examplebucket-1250000000</code>

The bucket-encryption command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--method	Specify the required operation, including put (configure bucket encryption), get (query bucket encryption), delete (delete bucket encryption).
None	--sse-algorithm	Encryption Algorithm (AES256, SM4, KMS)
None	--kms-master-key-id	KMS master key ID
None	--kms-algorithm	KMS Encryption Algorithm (AES256, SM4)

**Note:**

For more general options for this command (such as switching buckets or user accounts), see [Common Options](#).

## Configuring Bucket Encryption Policy

### Operation Example

Set the bucket encryption policy for the bucket named `example-alias`. The command is as follows:

```
./coscli bucket-encryption --method put cos://example-alias --sse-algorithm KMS
```

## Query Bucket Encryption Policy

### Operation Example

Query the encryption policy information of the bucket named `example-alias`.

```
./coscli bucket-encryption --method get cos://example-alias
```

Output the following result.

```

SECTION      | KEY          | VALUE
-----+-----+-----
Encryption  | Algorithm    | KMS
+          +-----+-----+
          | KMS Key ID  | *****-****-****-****-*****
+          +-----+-----+
          | Status      | Enabled
-----+-----+-----

COS Bucket Encryption Configuration

Encryption Details:
- Type: Server-Side Encryption with KMS-Managed Keys (SSE-KMS)
- Description: Tencent Cloud Key Management System (KMS) manages
encryption keys
- KMS Key ID: *****-****-****-****-*****
- Key Type: Customer Master Key (CMK)

```

## Deleting a Bucket Encryption Policy

### Operation Example

Delete the bucket encryption policy of the bucket named `example-alias`. The command is as follows:

```
./coscli bucket-encryption --method delete cos://example-alias
```

# Bucket Encryption Policy –bucket–encryption

Last updated: 2025-10-28 11:35:41

The bucket-policy command is used to set, query, and delete bucket policy.

## Note:

- To query bucket policy, when performing [authorization policy](#), set action to `cos:GetBucketPolicy`.
- To set a bucket policy, when performing [authorization policy](#), set action to `cos:PutBucketPolicy`.
- Note: To delete bucket policy, when performing [authorization policy](#), set action to `cos>DeleteBucketPolicy`.

For more authorization, please refer to [business APIs that support CAM](#).

## Command Syntax

```
./coscli bucket-policy --method [method] cos://<bucket-name>
```

The bucket-policy command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;</code>	Specify the target bucket. It supports the use of the bucket alias or bucket name in <a href="#">configuration parameters</a> . If you use the bucket name for access, you must carry the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias</code> Access with the bucket name: <code>cos://examplebucket-1250000000</code>

The bucket-policy command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--method	Specify the operation to perform, including put (set bucket policy), get (query bucket policy), delete (delete bucket policy)
None	--policy	Bucket policy (JSON format or directly specify the file path, such as /data/policy.txt, which needs to be specified as <code>file:///data/policy.txt</code> )

**Note:**

For more general options for this command (such as switching buckets or user accounts), see [Common Options](#).

## Setting a Bucket Policy

### Operation Example

Set the bucket policy for the bucket named `example-alias`. The command is as follows (see [PUT Bucket policy](#)):

```
./coscli bucket-policy --method put cos://example-alias --policy="{\"Statement\":[...]}"
```

## Querying a Bucket Policy

### Operation Example

Query the policy information for the bucket named `example-alias`.

```
./coscli bucket-policy --method get cos://example-alias
```

Output result, as shown below.

```
SECTION | KEY | VALUE
```

```

-----+-----+-----
-----
Policy          | Version   | 2.0
+-----+-----+-----
-----+
          |          |
+-----+-----+-----
-----+
Statement #1   | SID       | costs-12331231231123123-80285-3
+-----+-----+-----
-----+
          | Effect    | allow
+-----+-----+-----
-----+
          | Principal | qcs:
          |          | - qcs::cam::uin/1000000000:uin/1000000000
          |          | - qcs::cam::uin/1000000000:uin/1000000000
          |          |
+-----+-----+-----
-----+
          | Action    | name/cos:GetBucket
+-----+-----+-----
-----+
          | Resource  | qcs::cos:ap-nanjing:uid/1240000000:test-
1240000000/*
+-----+-----+-----
-----+
          | Condition | ip_equal:
          |          | qcs:ip:
          |          | - 11.9.10.8
          |          |
-----+-----+-----
-----
Bucket Policy Information

```

## Deleting a Bucket Policy

### Operation Example

Delete the bucket policy for the bucket named `example-alias`. The command is as follows:

```
./coscli bucket-policy --method delete cos://example-alias
```

# Inventory – inventory

Last updated: 2025-10-28 11:36:08

The inventory command is used to set, query, delete, list, and trigger a one-time list.

## Note:

- To query the inventory, when performing [authorization policy](#), set action to `cos:GetBucketInventory`.
- To set the inventory, when performing [authorization policy](#), set action to `cos:PutBucketInventory`.
- To delete the inventory, when performing [authorization policy](#), set action to `cos:DeleteBucketInventory`.
- To list the inventory, when performing [authorization policy](#), set action to `cos:GetBucketInventory`.
- To trigger a one-time list, when performing [authorization policy](#), set action to `cos:PostBucketInventory`.

For more authorization, please refer to [Business APIs Supporting CAM](#).

## Command Syntax

```
./coscli inventory --method [method] cos://<bucket-name>
```

The inventory command includes the following parameters:

Parameter Format	Description	Example
<code>cos://&lt;bucket-name&gt;</code>	Specify the target bucket. Supports using the bucket alias in <a href="#">configuration parameters</a> or the bucket name for access. If using the bucket name for access, you must also include the <code>endpoint</code> flag.	Access with the bucket alias: <code>cos://example-alias</code> Access with the bucket name: <code>cos://examplebucket-1250000000</code>

The inventory command includes the following optional flags:

Flag Abbreviation	Flag Name	Description
-h	--help	Views the usage of this command.
None	--method	Specify the required operation, including put (set bucket inventory), get (query bucket inventory), delete (delete bucket inventory), list (list bucket inventory), post (initiate one-time list).
None	--task-id	Inventory Task ID
None	--configuration	Inventory configuration (supports JSON format, XML format, or directly specifying a file path, such as /data/configuration.txt, which needs to be specified as <code>file:///data/configuration.txt</code> )

**Note:**

For more general options for this command (such as switching buckets or user accounts), see [Common Options](#).

## Setting Bucket Inventory

### Operation Example

Set the inventory for the bucket named example-alias. The command is as follows (specific format see [Setting Bucket Inventory](#)):

```
./coscli inventory --method put cos://example-alias --task-id list4 --configuration "<InventoryConfiguration>...</InventoryConfiguration>"
```

## Querying Bucket Inventory

### Operation Example

Query the manifest tasks of the bucket named example-alias list4.

```
./coscli inventory --method get cos://example-alias --task-id list4
```

Output the following result.

```

SECTION      | KEY                | VALUE
-----+-----+-----
Basic        | ID                 | list4
+-----+-----+-----
              | Enabled            | false
+-----+-----+-----
              | Included Versions | All
+-----+-----+-----
              |                    |
+-----+-----+-----
              |                    |
+-----+-----+-----
Schedule     | Frequency          | Weekly
+-----+-----+-----
              |                    |
+-----+-----+-----
              |                    |
+-----+-----+-----
Destination  | Bucket             | qcs::cos:ap-nanjing::test-
125000000
+-----+-----+-----
              | Format              | CSV
+-----+-----+-----
              | Account ID         | 100000000
+-----+-----+-----
              | Prefix              | list4
+-----+-----+-----
              |                    |
+-----+-----+-----
              |                    |
+-----+-----+-----
Filter       | Prefix              | myPrefix

```

```

+-----+-----+-----+
+-----+
+ | Tag | age=18 |
+-----+
+-----+
+ | Period | 2026-01-18 to 2019-09-17 |
+-----+
+-----+
+ | | |
+-----+
+-----+
Optional Fields | Field | Size
+-----+
+ | | |
+-----+
+ | | ETag |
+-----+
+ | | StorageClass |
+-----+
+ | | IsMultipartUploaded |
+-----+
+ | | Tag |
+-----+
+ | | LastModifiedDate |
+-----+
+-----+

```

Inventory Configuration Details

## Deleting a Bucket Inventory

### Operation Example

Delete the manifest task list4 of the bucket named example-alias. The command is as follows:

```
./coscli inventory --method delete cos://example-alias --task-id list4
```

## Listing All Bucket Inventories

## Operation Example

List all manifest tasks of the bucket named example-alias.

```
./coscli inventory --method list cos://example-alias
```

Output the following result.

```

  ID      | STATUS | SCHEDULE           | INCLUDED OBJECT VERSIONS |
DESTINATION           | FILTER
| FIELDS
-----+-----+-----+-----+-----
list1 | Enabled | Frequency: Daily | All |
Bucket: qcs::cos:ap-nanjing::test-125000000 | Period: 2019-09-
17 to 2019-09-17 | Size
|
Format: CSV |
| ETag
|
Account: 1000000000 |
| StorageClass
|
Prefix: list1 |
| IsMultipartUploaded
|
| ReplicationStatus
|
| LastModifiedDate
-----+-----+-----+-----+-----
list3 | Disabled | Frequency: Weekly | All |
Bucket: qcs::cos:ap-nanjing::test-125000000 | Period: 2026-01-
18 to 2019-09-17 | Size
|
Format: CSV |
| ETag

```

```

Account: 10000000000
| StorageClass
Prefix: list3
| IsMultipartUploaded
| ReplicationStatus
| LastModifiedDate
-----+-----+-----+-----+-----
-----+-----
list4 | Disabled | Frequency: Weekly | All
Bucket: qcs::cos:ap-nanjing::test-125000000 | Prefix: myPrefix
| Size
Format: CSV | Tags: age=18
| ETag
Account: 10000000000 | Period: 2026-01-
18 to 2019-09-17 | StorageClass
Prefix: list4
| IsMultipartUploaded
| Tag
| LastModifiedDate
-----+-----+-----+-----+-----
-----+-----
Detailed COS Bucket Inventory Configurations

Total inventory configurations: 3

```

## Initiating a One-Time List

### Operation Example

Initiate a one-time inventory task for the bucket named `example-alias`. The command is as follows (specific format see [Initiating a One-Time Inventory Task](#)):

```
./coscli inventory --method post cos://example-alias --task-id list4 --  
configuration "<InventoryConfiguration>...</InventoryConfiguration>"
```

# FAQs

Last updated: 2023-09-07 12:27:07

## What's the difference between COSCLI and COSCMD?

1. COSCLI is a binary package compiled using Golang. It is ready-to-use and does not require installing any dependency before the installation and deployment. However, COSCMD is compiled in Python and requires installing the Python environment and dependency packages.
2. COSCLI support setting bucket aliases, following users to use a short string instead of `<BucketName-APPID>` for convenience. COSCMD does not support bucket aliases, enquiring users to input `<BucketName-APPID>` to specify a bucket, making the commands more complex and less readable.
3. COSCLI support configuring multiple buckets in the configuration file and support cross-bucket operations. However, with COSCMD, you can only configure one bucket in the configuration file and the commands for cross-bucket operations are long.

# COS Migration

Last updated: 2024-11-20 19:49:17

## Note

COS Migration is an all-in-one tool that integrates the COS data migration feature. You can use it to migrate local data to COS through simple configurations and steps. It has the following features:

- **Checkpoint restart:** restarting uploads from checkpoints is supported. For large files, if the upload exits halfway or service failure occur, you can run the tool again to restart the upload.
- **Multipart upload:** An object can be upload to COS in multiple parts.
- **Parallel upload:** Multiple objects can be upload at the same time.
- **Migration verification:** Migrate objects can be verified.

### Note

- COS Migration only support UF-8 for encoding.
- If you use this tool to upload a file that already has the same name, the existing file will be overwrite. You need to configure the tool to skip files with the same name.
- For scenario other than local data migration, please prioritize using the [Migration Service Platform](#).
- COS Migration is used for **one-time** migration but is not suitable for continuous sync. For example, if files are added locally every day and need to be continuously synced to COS, then in order to avoid repeated migration tasks, COS Migration will save the records of successful migration. In case of continuous sync, the record scanning time will keep increasing. We recommend you use COSBrowser as described in [User Guide for Desktop Version](#) for this scenario.

## Operating Environment

### System environment

Windows, Linux, or macOS.

### Software requirements

- JDK 1.8 X64 or above. For more information, see [Java Installation and Configuration](#).
- IFUNC needs to be supported on Linux and the binutils version should be later than 2.20.

# How to Use

## 1. Obtain the tool

Download COS Migration [here](#).

## 2. Decompress the tool package

### Windows

Decompress the package and save it to a directory, for example:

```
C:\Users\Administrator\Downloads\cos_migrate
```

### Linux

Decompress the package and save it to a directory, for example:

```
unzip cos_migrate_tool_v5-master.zip && cd cos_migrate_tool_v5-master
```

## Migration tool structure

The structure of the properly decompressed COS Migration folder is as follow:

```
COS_Migrate_tool
|--conf # Directory of the configuration file
|   |--config.ini # Migration configuration file
|--db # Record of storage migration successes
|--dep #Main logic compiled JAR package
|--log # Log generated during tool execution
|--opbin # Script for compiling
|--result #Directory for saving successful migration records, with the
file name format "date.out" and the record format "absolute path\tfile
size\tlast modified time"
|--src # Source code of the tool
|--tmp # Temporary file storage directory
|--.gitignore # Files and folders ignored by Git version control
|--pom.xml # Project configuration file
|--README # Readme document
|--start_migrate.sh # Migration startup script for Linux
|--start_migrate.bat # Migration startup script for Windows
```

**Note**

- The db directory mainly records the IDs of files successfully migrated by the tool. For each migration task, the IDs in db will be compared first, and if the ID of the current file has already been recorded, the current file will be skipped; otherwise, it will be migrated.
- The log directory keeps all the logs generated during migration. If an error occurs, please check the error.log in this directory first.

### 3. Modify the config.ini configuration file

Before running the migration startup script, you need to modify the config.ini file (path: `./conf/config.ini`) first. This file contains the following parts:

#### 3.1 Configure the migration type

The "type" represents the migration type, and users should fill in the corresponding identifier based on their migration requirements. For example, if you need to migrate local data to COS, the configuration content for `[migrateType]` is `type=migrateLocal`.

```
[migrateType]
type=migrateLocal
```

Currently supported migration types include:

migrateType	Description
migrateLocal	From local system to COS

#### 3.2 Configure the migration task

You can configure a migration task based on your actual migration needs. Main configuration items include the destination COS bucket and task properties.

```
# The public configuration section of the migration tool contains the
account information for the target COS to which the data will be
migrated.
[common]
secretId=COS_SECRETID
secretKey=COS_SECRETKEY
bucketName=examplebucket-1250000000
region=ap-guangzhou
```

```
storageClass=Standard
cosPath=/
https=off
tmpFolder=./tmp
smallFileThreshold=5242880
smallFileExecutorNum=64
bigFileExecutorNum=8
entireFileMd5Attached=on
daemonMode=off
daemonModeInterVal=60
executeTimeWindow=00:00,24:00
outputFinishedFileFolder=./result
resume=false
skipSamePath=false
```

Name	Description	Default value
secretId	SecretId of your key. Replace <code>COS_SECRETID</code> with your real key information, which can be obtained on the TencentCloud API key page in the <a href="#">CAM Console</a>	-
secretKey	SecretKey of your key. Replace <code>COS_SECRETKEY</code> with your real key information, which can be obtained on the TencentCloud API key page in the <a href="#">CAM Console</a>	-
bucketName	Destination bucket name: The naming format is <code>&lt;BucketName-APPID&gt;</code> , meaning the bucket name must include the APPID, such as <code>examplebucket-1250000000</code> .	-
region	Region information of the destination bucket. For the abbreviations for COS regions, see <a href="#">Regions and Access Endpoints</a> .	-
storageClass	After data migration, the available storage types are Standard (STANDARD storage), Standard_IA (IA storage), Archive (ARCHIVE storage), Maz_Standard (Multi-AZ STANDARD storage), and Maz_Standard_IA (Multi-AZ IA storage). For more information, see <a href="#">Storage Class Overview</a> .	Standard
cosPath	The COS path to migrate to. <code>/</code> represents migrating to the root path of the Bucket, while <code>/folder/doc/</code> indicates migrating to the <code>/folder/doc/</code> directory within the Bucket. If <code>/folder/doc/</code> does not exist, the path will be created automatically.	/
https	Whether to transfer via HTTPS. on: Yes, off: No. It takes time to enable transfer via HTTPS, which is suitable for scenarios that demand high security.	off
tmpFolder	During the migration process from other cloud storage to COS, a temporary file directory is used to store intermediate files, which will be deleted after the migration is complete. The required format is an absolute path: In Linux, the separator is a single forward slash, for example, <code>/a/b/c</code> . In Windows, the separator is two backslashes, for example: <code>E:\\a\\b\\c</code> By default, it is the tmp directory in the tool's location	./tmp

	By default, it is the tmp directory in the tool's location path.	
smallFileThreshold	Number of bytes as the threshold for small files. If the data size is higher than or equal to this threshold, multipart upload is used; otherwise, simple upload is used. The default value is 5 MB	5242880
smallFileExecutorNum	Concurrency for uploading small files (smaller than smallFileThreshold) via simple upload. Decrease the concurrency if files are uploaded to COS via public network with low bandwidth	64
bigFileExecutorNum	Concurrency for uploading large files (greater than or equal to smallFileThreshold) via multipart upload. Decrease the concurrency if files are uploaded to COS via public network with low bandwidth	8
entireFileMd5Attached	This indicates whether the migration tool calculates the MD5 of the entire file and stores it in the custom header "x-cos-meta-md5" of the file for subsequent verification, because the etag of a large file uploaded to COS in multiple parts is not the MD5 of the entire file	on
daemonMode	Whether to enable daemon mode. on: yes; off: no. In daemon mode, the program will keep performing synchronization. The synchronization interval is configured by the daemonModeInterVal parameter	off
daemonModeInterVal	This indicates the interval in seconds between two rounds of synchronization	60
executeTimeWindow	Execution time window: The time granularity is in minutes, and this parameter defines the time period during which the migration tool runs daily. For example: Parameters 03:30, 21:00 indicate that the task will be executed between 03:30 AM and 09:00 PM. During other times, the task will enter a sleep state, pausing the migration while preserving its progress until the next time window when it will automatically resume. Note that the latter time must be greater than the former time.	00:00,24:00
outputFinishedFileFolder	This directory stores the successful migration results. The result files are named by date, such as <code>./result/2021-05-27.out</code> , where <code>./result</code> is the created directory. Each line in the file content is formatted as: absolute path\tfile size\tlast modified time. If set to empty, no results will be output.	./result

<code>resume</code>	Whether to continue with the result of the last run and traverse through the list of files from the source. The tool starts from scratch by default.	<code>false</code>
<code>skipSamePath</code>	Whether to skip the current file if a file with the same name already exists in COS. By default, the tool does not skip the current file: it overwrites the existing file.	<code>false</code>
<code>requestTryCount</code>	Total number of attempts for each file upload.	<code>5</code>

### 3.3 Configure the data source

Configure the corresponding sections based on the `[migrateType]` migration type. For example, if the configuration content of `[migrateType]` is `type=migrateLocal`, the user only needs to configure the `[migrateLocal]` section.

#### 3.3.1 Configure a local data source "migrateLocal"

If you are migrating from a local system to COS, configure this section. The specific configuration items and descriptions are as follows:

```
# Configuration for Migrating from Local to COS
[migrateLocal]
localPath=E:\\code\\java\\workspace\\cos_migrate_tool\\test_data
excludes=
ignoreModifiedTimeLessThanSeconds=
```

Configuration items	Description
localPath	<p>Local directory, required in absolute path format:</p> <ul style="list-style-type: none"> <li>In Linux, the separator is a single forward slash, for example, <code>/a/b/c</code>.</li> <li>In Windows, the separator is two backslashes, for example: <code>E:\\a\\b\\c</code>.</li> </ul> <p>Caution: Only the directory path can be entered for this parameter, not the specific file path. Otherwise, it will cause an error in parsing the target object name. In the case of <code>cosPath=/</code>, it will be incorrectly parsed as a bucket creation request.</p>
excludes	<p>Absolute path of the directory or file to be excluded, which means that some directories or files in the <code>localPath</code> are not to be migrated. Multiple absolute paths are separated by semicolons. If this is left blank, all files in the <code>localPath</code> are to be migrated.</p>
ignoreModifiedTimeLessThanSeconds	<p>Exclude files that have an update time less than a certain period of time from the current time (in seconds). This item is left blank by default, indicating files are not to be filtered by the time specified by <code>lastmodified</code>. It is suitable for scenarios where you run the migration tool while updating files and don't want files being updated to be migrated to COS. For example, if it is configured as <code>300</code>, only files updated at least 5 minutes ago will be uploaded.</p>

## 4. Run the migration tool

### Windows

Double-click `start_migrate.bat` to run it.

### Linux

1. Read the configuration from the `config.ini` file. The run command is:

```
sh start_migrate.sh
```

2. Read the configuration of some parameters from the command line. The run command is:

```
sh start_migrate.sh -Dcommon.cosPath=/savepoint0403_10/
```

#### Note

- The tool supports reading configuration items in two ways: command line or configuration file.
- The command line takes precedence over the configuration file, i.e., for the same configuration item, the parameter in the command line takes precedence.
- Using configuration options in the command line allows users to run different migration tasks simultaneously, provided that the key configuration items in the two tasks are not identical, such as bucket names, COS paths, and source paths to be migrated. Different migration tasks write to different db directories, ensuring concurrent migration. Please refer to the db information in the tool structure mentioned earlier.
- The configuration item format is `-D{sectionName}.{sectionKey}={sectionValue}`, where `sectionName` is the section name in the configuration file, `sectionKey` represents the configuration item name within the section, and `sectionValue` represents the configuration item value within the section. For example, to set the COS path for migration, use `-Dcommon.cosPath=/bbb/ddd`.

## Migration Mechanism and Process

### How Migration Works

COS Migration has a status. Successful migrations will be recorded in the format of `KV` in the `leveldb` file under the `db` directory. Before each migration, check whether the path to which data is migrated has been recorded in the `db` directory. If yes, and its attribute is the same as that in `db`, the migration will be skipped; otherwise, the migration will be executed. The attribute varies by migration type. For local migration, `mtime` determines whether to migrate. For migration from other cloud storage services and bucket replication, the etag and length of the source file determine whether to migrate. Therefore, we search for records of successful migrations in the `db` directory rather than in COS. If a file is deleted or modified via COSCMD or the console rather than the migration tool, the migration tool cannot detect this change, and the file will not be re-migrated.

### Migration Steps

1. The configuration file is read, the corresponding configuration section is read according to the migration type, and parameters are checked.
2. The IDs of the files to be migrated are scanned for and compared in the db directory according to the migration type to decide whether upload is allowed.
3. During the migration process, the execution results will be printed, where "inprogress" indicates ongoing migration, "skip" means skipped, "fail" denotes failure, "ok" signifies success, and "condition\_not\_match" represents files skipped due to not meeting migration

conditions (such as lastmodified and excludes). Detailed information about failures can be found in the error logs. The execution process is illustrated in the following diagram:

```

chengwu@vm-24-72-centos[10:25:10]:~/code/java/cos_migrate_tool
$ sh start_migrate.sh
[skip] task info: [taskType: migrateAws] [bucket: chengwus3sdkgz-1251668577] [cosPath: /aws0403_17/aaa/bbbslash.txt]
[DownloadOk] [key: slash.txt] [byteDownload/ byteTotal/ percentage: 6/ 6/ 100.00%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 10240/ 209715200/ 0.00%]
[UploadInProgress] [key: /aws0403_17/slash.txt] [byteSent/ byteTotal/ percentage: 6/ 6/ 100.00%]
[ok] task info: [taskType: migrateAws] [bucket: chengwus3sdkgz-1251668577] [cosPath: /aws0403_17/slash.txt]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 10240/ 209715200/ 0.00%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 236126/ 209715200/ 0.11%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 392798/ 209715200/ 0.19%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 514654/ 209715200/ 0.25%]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 96863/ 209715200/ 0.05%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 688734/ 209715200/ 0.33%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 827998/ 209715200/ 0.39%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 949854/ 209715200/ 0.45%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1089118/ 209715200/ 0.52%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1228382/ 209715200/ 0.59%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1315422/ 209715200/ 0.63%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1385054/ 209715200/ 0.66%]
[DownloadInProgress] [key: aws_200M.txt.copy] [byteDownload/ byteTotal/ percentage: 166495/ 209715200/ 0.08%]
[DownloadInProgress] [key: aws_200M.txt] [byteDownload/ byteTotal/ percentage: 1506910/ 209715200/ 0.72%]

```

4. Upon completion of the entire migration, statistical information will be printed, including the cumulative number of successful migrations, failures, skipped files, and time consumed. For failed cases, please check the error log or rerun the migration tool, as it will skip successfully migrated files and retry the unsuccessful ones. The result illustration is shown in the following figure:

```

migrateAli over! op statistics:
      op_status : ALL_OK
      migrate_ok : 530038
      migrate_fail : 0
      migrate_skip : 496264
      start_time : 2018-03-19 15:52:02
      end_time : 2018-03-19 16:54:38
      used_time : 3756 s

```

## FAQs

If you encounter migration failures, runtime errors, or other abnormal situations while using the COS Migration tool, please refer to [Common Issues with COS Migration Tool](#) for solutions.

## Summary

Of course, COS also offers the above applications and services but also provides a variety of popular open-source applications integrated with Tencent Cloud COS plugins. Click [here](#) to launch and start using them immediately!

# FTP Server

Last updated: 2023-09-07 12:27:41

## Feature Overview

COS FTP Server allows you to directly operate on COS objects and directories over FTP protocol, including uploading/downloading/deleting files and creating folders. This tool is developed with Python, which makes the installation easier.

## Note

**Upload Mechanism:** Stream-based upload without saving to local disk. Simply configure the working directory according to the standard FTP protocol, without occupying actual disk storage space.

**Download Mechanism:** Directly stream data back to the client.

**Directory Mechanism:** The bucket serves as the root directory for the entire FTP Server, and multiple subdirectories can be created under the bucket.

**Multi-bucket Binding:** Supports binding to multiple buckets simultaneously.

### Note

**Multi-bucket Binding:** This is achieved by using different FTP Server working paths (`home_dir`). Therefore, when specifying different buckets and user information, you must ensure that the `home_dir` is unique.

**Delete Operation Restriction:** In the new FTP Server, you can configure the `delete_enable` option for each FTP user to indicate whether the user is allowed to delete files.

**Supported FTP Commands:** `put`, `mput`, `get`, `rename`, `delete`, `mkdir`, `ls`, `cd`, `bye`, `quit`, `size`.

**Unsupported FTP Commands:** `append`, `mget` (native `mget` command is not supported, but batch downloading is still possible in some Windows clients, such as FileZilla client.)

### Note

The FTP Server tool currently does not support checkpoint restart functionality.

## Getting Started

### System environment

- OS: Linux. The [CVM](#) of Tencent CentOS series is recommended. Windows systems are not supported for now.

- **psutil-dependent Linux package:** `python-devel` or `python-dev`, depending on the names of different Linux distributions. It is added using Linux package manager, such as `yum install python-devel` or `aptitude install python-dev`.
- **Python interpreter version:** Python 2.7. For more information on the installation and configuration, see [Python](#).

**Note**

FTP Server tool does not support Python 3.

- **Dependent packages:**
- [cos-python-sdk-v5](#) ( $\geq 1.6.5$ )
- [pyftplib](#) ( $\geq 1.5.2$ )
- [psutil](#) ( $\geq 5.6.1$ )

## Usage Limits

Applicable to COS XML version.

## Installation and Operation

Download the FTP Server tool at [cos-ftp-server](#). The installation steps are as follows:

1. Enter the FTP Server directory, and run `setup.py` to install FTP Server and its dependent libraries (network required).

```
python setup.py install # Your account should use sudo or have the
root permission.
```

2. Copy the configuration sample file `conf/vsftpd.conf.example` and name it `conf/vsftpd.conf`. See [Configuration File](#) of this document to correctly configure bucket and user information.
3. Run `ftp_server.py` to start FTP Server.

```
python ftp_server.py
```

FTP Server can also be started in the following two ways:

- Execute the `nohup` command to start it in the backend process:

```
nohup python ftp_server.py >> /dev/null 2>&1 &
```

- Run the command in the background using the screen command (requires the installation of the screen tool):

```
screen -dmS ftp
screen -r ftp
python ftp_server.py
#Use the keyboard shortcut Ctrl + A + D to go back to the main screen.
```

## Stop Operation

- If you are running the FTP Server directly or using the screen method to run it in the background, you can use the shortcut key `Ctrl+C` to stop the FTP Server.
- If FTP Server is started with the `nohup` command, you can stop it by the following way:

```
ps -ef | grep python | grep ftp_server.py | grep -v grep | awk '{print $2}' | xargs -I{} kill {}
```

## Configuration File

The configuration sample file for FTP Server is `conf/vsftpd.conf.example`. Copy and name it `vsftpd.conf`, and then configure it as follows:

```
[COS_ACCOUNT_0]
cos_secretid = COS_SECRETID      # Replaced with your SECRETID
cos_secretkey = COS_SECRETKEY    # Replaced with your SECRETKEY
cos_bucket = examplebucket-1250000000
cos_region = region            # Replaced with your bucket region
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user0          # Replaced with the local path you want the
FTP to mount to (which should be an actual existing path but not a soft
link)
ftp_login_user_name=user0       # Replaced with a custom username
ftp_login_user_password=pass0   # Replaced with a custom password
authority=RW                    # The user's read and write permissions. R:
read; W: write; RW: both.
delete_enable=true             # true allows the FTP user to delete files by
default; false prohibits the user to delete files.
```

```
[COS_ACCOUNT_1]
cos_secretid = COS_SECRETID      # Replaced with your SECRETID
cos_secretkey = COS_SECRETKEY    # Replaced with your SECRETKEY
cos_bucket = examplebucket-1250000000
cos_region = region             # Replaced with your bucket region
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user1          # Replaced with the local path you want the
FTP to mount to (which should be an existing path but not a soft link)
ftp_login_user_name=user1       # Replaced with a custom username
ftp_login_user_password=pass1   # Replace with a user-defined password
authority=RW                     # The user's read and write permissions. R:
read; W: write; RW: both.
delete_enable=false            # true allows the FTP user to delete files by
default; false prohibits the user to delete files.

[NETWORK]
# If the FTP Server is behind a gateway or NAT, you can use this
configuration item to specify the IP address or domain name of the
gateway for FTP.
masquerade_address = XXX.XXX.XXX.XXX
# The default listening port for the FTP Server is 2121. Ensure that the
firewall allows traffic through this port (e.g., if you deploy the FTP
Server tool on a Tencent Cloud CVM, you need to open this port in the
CVM security group).
listen_port = 2121
# passive_port allows you to set the port range for passive mode, which
defaults to the range [60000, 65535]. Be sure to open these ports in the
firewall (e.g., CVM security group).
passive_port = 60000,65535

[FILE_OPTION]
# The default maximum single file size is up to 200 GB; it is not
recommended to set it too large.
single_file_max_size = 21474836480

[OPTIONAL]
# For the following settings, it is recommended to keep the default
values unless there is a specific need. If you need to set a value,
```

```
please enter a reasonable integer.
min_part_size      = default
upload_thread_num  = default
max_connection_num = 512
max_list_file      = 10000          # The maximum number of files
to be listed by ls command. It is not recommended to go beyond this
limit. Otherwise, high latency of ls command will occur.
log_level          = INFO           # Set the log output level.
log_dir            = log            # Set the directory to store
logs. Default: log under the FTP Server directory.
```

### Note

- To bind each user to a different bucket, simply add a [COS\_ACCOUNT\_X] section. For each different COS\_ACCOUNT\_X section, the following explanations apply:
  - The username ( `ftp_login_user_name` ) and the home directory ( `home_dir` ) under each account must be unique, and the home directory must be a directory that exists in the system.
  - The number of users logging in to each COS FTP Server simultaneously cannot exceed 100.
  - `endpoint` and `region` will not take effect at the same time. To use the public cloud COS service, enter the `region` field correctly. The `endpoint` is commonly used in the privatized deployment environment. When both '`region`' and '`endpoint`' are entered, '`endpoint`' will take precedence.
- The OPTIONAL settings in the configuration file are provided for advanced users to adjust upload performance. By reasonably adjusting the size of upload parts and the number of concurrent upload threads according to the machine's performance, better upload speeds can be achieved. General users do not need to make adjustments and can keep the default values. Additionally, there is an option to limit the maximum number of connections. If you do not want to limit the maximum number of connections, you can enter 0, which means there is no restriction on the maximum number of connections (however, you should evaluate this based on your machine's performance).
- In the `masquerade_address` configuration item of the configuration file, it is generally recommended to specify the IP address used by the client to connect to the COS FTP Server. If you have any questions, please refer to the [FTP Server Tool Frequently Asked Questions](#) document.
  - Assume that the FTP Server has more than one IP address, and after running the `ifconfig` command, you get a private ENI IP `10.xxx.xxx.xxx`, which is

mapped to the public IP `119.xxx.xxx.xxx`. At this time, if the FTP Server does not explicitly set `masquerade_address` to the public IP (119.xxx.xxx.xxx) that the client uses to access the server, the FTP Server in Passive mode may use the private IP (10.xxx.xxx.xxx) to return packets to the client. As a result, the client is able to connect to the FTP Server, but cannot return data packets to the client properly. Therefore, generally speaking, we recommend you to set `masquerade_address` to the IP address that your client is using to connect to the Server.

- In the configuration file, the `listen_port` option is the listening port for the COS FTP Server, with a default value of 2121. The `passive_port` option is the data channel listening port range for the COS FTP Server, which defaults to a range between 60000 and 65535. When connecting to the COS FTP Server, ensure that the firewall allows traffic through the ports specified in `listen_port` and `passive_port`.

## Quick Practice

### Accessing COS FTP Server using Linux `ftp` command

1. Install the Linux FTP client.

```
yum install -y ftp
```

2. Open the Linux command line, and use the command `ftp [ip address] [port No.]` to connect to the COS FTP Server. Example:

```
ftp 192.xxx.xx.103 2121
```

- In the FTP command, the IP setting corresponds to the `masquerade_address` configuration item in the example configuration file `conf/vsftpd.conf.example`. In this case, the IP is set to 192.xxx.xx.103.
  - In the FTP command, the port setting corresponds to the `listen_port` configuration item in the example configuration file `conf/vsftpd.conf.example`. In this case, it is set to 2121.
3. After running the above command, you will see **Name** and **Password** input fields. Enter the values configured in the COS FTP Server configuration items `ftp_login_user_name` and `ftp_login_user_password` to establish a successful connection.
    - **Name:** Corresponds to the `ftp_login_user_name` configuration item in the example configuration file `conf/vsftpd.conf.example` (requires configuration).
    - **Password:** Corresponds to the `ftp_login_user_password` configuration item in the example configuration file `conf/vsftpd.conf.example` (requires configuration).

## Accessing COS FTP Server using FileZilla

1. Download and install [FileZilla client](#).
2. After configuring the access information for COS FTP Server on your FileZilla client, click **Quick Connect**.
  - **Host (H):** Corresponds to the **masquerade\_address** configuration item in the example configuration file `conf/vsftpd.conf.example`. In this example, the IP is set to `192.xxx.xx.103`.

### Note

If the COS FTP Server is behind a gateway or NAT, you can use this section to specify the gateway's IP address or domain name as the Server' s IP address.

- **Username (U):** Corresponds to the **ftp\_login\_user\_name** configuration item in the example configuration file `conf/vsftpd.conf.example` (requires configuration).
- **Password (W):** Corresponds to the **ftp\_login\_user\_password** configuration item in the example configuration file `conf/vsftpd.conf.example` (requires configuration).
- **Port (P):** Corresponds to the **listen\_port** configuration item in the example configuration file `conf/vsftpd.conf.example` . In this case, it is set to 2121.

## FAQs

If any error occurs or you have any question on the upload limit while using FTP Server, see [FTP Server FAQs](#).

# Hadoop

Last updated: 2025-10-15 16:45:07

## Note

Hadoop-COS implements a standard Hadoop file system on the Tencent Cloud COS platform. It helps integrate COS with big data computing frameworks such as Hadoop, Spark, and Tez, so that they can read and write COS data as they do with HDFS file systems. Hadoop-COS uses `cosn` as the URI scheme, so it is also called CosN file system.

## Operating Environment

### System environment

Windows, Linux, or macOS.

### Software requirements

Hadoop-2.6.0 or later.

#### Note:

- Hadoop-COS has been integrated into Apache Hadoop 3.3.0. For more information, see [Integration of Tencent COS in Hadoop](#).
- If your version is earlier than Hadoop 3.3.0, or CDH has integrated the Hadoop-COS JAR package, you need to restart NodeManager to load the JAR package.
- To build a JAR package of a specified Hadoop version, modify `hadoop.version` in the `pom` file.

## Download and Installation

### Obtaining the Hadoop-COS JAR package and dependencies

Download link: [Hadoop-COS release](#)

### Install `hadoop-cos` plugin

1. Copy `hadoop-cos-{hadoop.version}-{version}.jar` and `cos_api-bundle-{version}.jar` to `$HADOOP_HOME/share/hadoop/tools/lib`.

#### Note:

Choose the appropriate JAR package based on the specific Hadoop version. If there is no matching JAR package in the release, you can modify the Hadoop version

number in the POM file and recompile it.

2. Navigate to the `$HADOOP_HOME/etc/hadoop` directory and edit the `hadoop-env.sh` file. Add the following content to include the COSN-related JAR files in the Hadoop environment variables:

```
for f in $HADOOP_HOME/share/hadoop/tools/lib/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
    export HADOOP_CLASSPATH=$f
  fi
done
```

## Configuration Method

### Configure HADOOP

Modify `$HADOOP_HOME/etc/hadoop/core-site.xml` by adding the information of COS users and implementation classes as shown below:

#### Note:

When modifying configuration items, you can directly modify the configuration items in the `core-site.xml` file of COSN. If you dynamically load Hadoop configuration through Spark, you need to add the `spark.hadoop.*` prefix to the configuration items, such as `spark.hadoop.fs.cosn.upload.buffer`. This configuration item can be set via `hadoopConfiguration` in `SparkContext` or specified in the shell using `spark-sql --conf`.

```
<configuration>
  <property>
    <name>fs.cosn.credentials.provider</name>

<value>org.apache.hadoop.fs.auth.SimpleCredentialProvider</value>
    <description>

        This option allows the user to specify how to get the
        credentials.

        Comma-separated class names of credential provider classes
        which implement
```

```
com.qcloud.cos.auth.COSCredentialsProvider:
```

```
1.org.apache.hadoop.fs.auth.SessionCredentialProvider:
```

Obtain the secret id and secret key from the URI:

```
cosn://secretId:secretKey@examplebucket-1250000000/;
```

```
2.org.apache.hadoop.fs.auth.SimpleCredentialProvider: Obtain
the secret id and secret key
```

```
from fs.cosn.userinfo.secretId and
fs.cosn.userinfo.secretKey in core-site.xml;
```

```
3.org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider:
```

Obtain the secret id and secret key

```
from system environment variables named COS_SECRET_ID and
COS_SECRET_KEY.
```

If unspecified, the default order of credential providers is:

```
1. org.apache.hadoop.fs.auth.SessionCredentialProvider
```

```
2. org.apache.hadoop.fs.auth.SimpleCredentialProvider
```

```
3.
```

```
org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider
```

```
4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider
```

```
5. org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider
```

```
6. org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider
```

```
7. org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider
```

```
</description>
```

```
</property>
```

```
<property>
```

```
<name>fs.cosn.userinfo.secretId</name>
```

```
<value>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</value>
```

```
<description>Tencent Cloud Secret Id</description>
```

```
</property>
```

```
<property>
```

```
<name>fs.cosn.userinfo.secretKey</name>
```

```
<value>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx</value>
```

```
<description>Tencent Cloud Secret Key</description>
```

```
</property>
```

```
<property>
  <name>fs.cosn.bucket.region</name>
  <value>ap-xxx</value>
  <description>The region where the bucket is located.
</description>
</property>

<property>
  <name>fs.cosn.bucket.endpoint_suffix</name>
  <value>cos.ap-xxx.myqcloud.com</value>
  <description>
    COS endpoint to connect to.
    For public cloud users, it is recommended not to set this
option, and only the correct area field is required.
  </description>
</property>

<property>
  <name>fs.cosn.impl</name>
  <value>org.apache.hadoop.fs.CosFileSystem</value>
  <description>The implementation class of the CosN Filesystem.
</description>
</property>

<property>
  <name>fs.AbstractFileSystem.cosn.impl</name>
  <value>org.apache.hadoop.fs.CosN</value>
  <description>The implementation class of the CosN
AbstractFileSystem.</description>
</property>

<property>
  <name>fs.cosn.tmp.dir</name>
  <value>/tmp/hadoop_cos</value>
  <description>Temporary files will be placed here.</description>
</property>

<property>
  <name>fs.cosn.upload.buffer</name>
  <value>mapped_disk</value>
```

```
<description>The type of upload buffer. Available values:
non_direct_memory, direct_memory, mapped_disk</description>
</property>

<property>
  <name>fs.cosn.upload.buffer.size</name>
  <value>134217728</value>
  <description>The total size of the upload buffer pool. -1 means
unlimited.</description>
</property>

<property>
  <name>fs.cosn.upload.part.size</name>
  <value>8388608</value>
  <description>Block size to use cosn filesystem, which is the
part size for MultipartUpload.
  Considering the COS supports up to 10000 blocks, user should
estimate the maximum size of a single file.
  For example, 8MB part size can allow writing a 78GB single
file.</description>
</property>

<property>
  <name>fs.cosn.maxRetries</name>
  <value>3</value>
  <description>
The maximum number of retries for reading or writing files to
COS, before we signal failure to the application.
  </description>
</property>

<property>
  <name>fs.cosn.retry.interval.seconds</name>
  <value>3</value>
  <description>The number of seconds to sleep between each COS
retry.</description>
</property>

<property>
  <name>fs.cosn.server-side-encryption.algorithm</name>
```

```

    <value></value>
    <description>The server side encryption algorithm.</description>
  </property>

  <property>
    <name>fs.cosn.server-side-encryption.key</name>
    <value></value>
    <description>The SSE-C server side encryption key.</description>
  </property>

  <property>
    <name>fs.cosn.client-side-encryption.enabled</name>
    <value></value>
    <description>Enable or disable the client encryption
function</description>
  </property>

  <property>
    <name>fs.cosn.client-side-encryption.public.key.path</name>
    <value>/xxx/xxx.key</value>
    <description>The direct path to the public key</description>
  </property>

  <property>
    <name>fs.cosn.client-side-encryption.private.key.path</name>
    <value>/xxx/xxx.key</value>
    <description>The direct path to the private key</description>
  </property>

</configuration>

```

It is not recommended to configure `fs.defaultFS` in production environments. If required for specific testing scenarios (such as `hive-testbench`, etc.), ensure that only one `fs.defaultFS` is configured. The following configuration information can be added:

```

<property>
  <name>fs.defaultFS</name>
  <value>cosn://examplebucket-1250000000</value>
  <description>

```

```
This option is not advice to config, this only used for  
some special test cases.
```

```
</description>  
</property>
```

## Configuration Items

Attribute Key	Note	Default value	Required
fs.cosn.userinfo.secretId/secretKey	Specifies the API key for your account. Log in to the <a href="#">CAM console</a> to view the key.	-	Required
fs.cosn.credentials.provider	<p>Methods to obtain SecretId and SecretKey are as follows:</p> <ol style="list-style-type: none"> <li>1. org.apache.hadoop.fs.auth.SessionCredentialProvider: Retrieves the secret id and secret key from the request URI. The format is: <code>cosn:// {secretId}:{secretKey} @examplebucket-1250000 000/.</code></li> <li>2. org.apache.hadoop.fs.auth.SimpleCredentialProvider: Obtain SecretId and SecretKey by reading fs.cosn.userinfo.secretId and fs.cosn.userinfo.secretKey from the core-site.xml configuration file.</li> <li>3. org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider: Obtain from system environment variables COS_SECRET_ID and COS_SECRET_KEY.</li> <li>4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider: Access using <a href="#">temporary key format</a>.</li> <li>5. org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider: Utilizes</li> </ol>	<p>If this configuration item is not specified, the following order will be used by default:</p> <ol style="list-style-type: none"> <li>1. org.apache.hadoop.fs.auth.SessionCredentialProvider</li> <li>2. org.apache.hadoop.fs.auth.SimpleCredentialProvider.</li> <li>3. org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider</li> <li>4. org.apache.hadoop.fs.auth.SessionTokenCredentialPr</li> </ol>	Not required

	<p>provider. Utilizes the role bound to Tencent Cloud CVM instances to obtain temporary keys for accessing COS.</p> <p>6. org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider: Utilizes the role bound to Tencent Cloud's Blackstone Physical Machine (CPM) to obtain temporary keys for accessing COS.</p> <p>7. org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider: Utilizes the role bound to a Tencent Cloud EMR instance to obtain temporary credentials for accessing COS.</p> <p>8. The org.apache.hadoop.fs.auth.RangerCredentialsProvider utilizes Ranger for obtaining the necessary credentials.</p>	<p>provider.</p> <p>5. org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider.</p> <p>6. org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider.</p> <p>7. org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider.</p>	
fs.cosn.useHttps	Indicates whether to use HTTPS as the transfer protocol for the COS server.	true	Not required
fs.cosn.impl	COSN implementation class for FileSystem; fixed as <code>org.apache.hadoop.fs.CosFileSystem</code> .	-	Required
fs.AbstractFileSystem.cosn.impl	The COSN implementation class for <code>AbstractFileSystem</code> , which is fixed at <code>org.apache.hadoop.fs.CosN</code> .	-	Required
	Please enter the region		

fs.cosn.bucket.region	information for the bucket you want to access. For a list of region abbreviations, refer to <a href="#">Regions and Access Domain Names</a> . For example: ap-beijing, ap-guangzhou, etc. Compatible with the original configuration: fs.cosn.userinfo.region.	-	Required
fs.cosn.bucket.endpoint_suffix	Specify the COS endpoint to connect to; this is an optional field. For public cloud COS users, Simply fill in the region configuration correctly. It is compatible with the original configuration: fs.cosn.userinfo.endpoint_suffix. When configuring this item, please delete the fs.cosn.bucket.region configuration item endpoint to take effect.	-	Not required
fs.cosn.tmp.dir	Sets a directory that actually exists. Temporary files generated in the running process will be stored here.	/tmp/hadoop_coss	Not required
fs.cosn.upload.part.size	The size of each block in the COSN file system is also the size of each part in a multipart upload. Since COS supports a maximum of 10,000 parts for multipart uploads, it is essential to estimate the largest possible single file size that may be used. For instance, when the part size is 8 MB, it can support a maximum single file upload of 78 GB. The	8388608 ( 8MB )	Not required

	maximum part size can be up to 2 GB, allowing for a maximum single file upload of 19 TB.		
<code>fs.cosn.upload.buffer</code>	The type of buffer used when uploading files with COSN. Currently, there are three types of buffers: <code>non_direct_memory</code> , <code>direct_memory</code> and <code>mapped_disk</code> . The <code>non_direct_memory</code> buffer uses JVM on-heap memory, the <code>direct_memory</code> buffer uses off-heap memory, while the <code>mapped_disk</code> buffer works based on memory file mapping.	<code>mapped_disk</code>	Not required
<code>fs.cosn.upload.buffer.size</code>	The size of buffer used during upload through COSN. A value of <code>-1</code> means unlimited. You can specify this value only if you set the buffer type to <code>mapped_disk</code> . If you specify a value greater than 0, it cannot be smaller than the block size. This parameter is compatible with the old parameter <code>fs.cosn.buffer.size</code> .	<code>-1</code>	Not required
<code>fs.cosn.block.size</code>	The size of a block in the COSN file system.	134217728 (128MB)	Not required
<code>fs.cosn.upload_thread_pool</code>	Specifies the number of concurrent threads when files are uploaded to COS via streams.	10	Not required
<code>fs.cosn.copy_thread_pool</code>	Specifies the number of threads used to copy and delete concurrent files during directory	3	Not required

	replication.		
fs.cosn.read.ahead.block.size	The size of a read-ahead block.	1048576 ( 1MB )	Not required
fs.cosn.read.ahead.queue.size	The length of the read-ahead queue.	8	Not required
fs.cosn.maxRetries	The maximum number of retries if an error occurs when accessing COS.	200	Not required
fs.cosn.retry.interval.seconds	Time interval between retries in seconds	3	Not required
fs.cosn.server-side-encryption.algorithm	Configure COS server-side encryption algorithms, supporting SSE-C, SSE-COS, and SSE-KMS. The default setting is empty (no encryption).	-	Not required
fs.cosn.server-side-encryption.key	<ul style="list-style-type: none"> <li>When enabling COS's SSE-C server-side encryption algorithm, you must configure the SSE-C encryption key, which should be a base64-encoded AES-256 key. By default, this field remains empty, indicating no encryption is applied.</li> <li>When activating COS's SSE-KMS server-side encryption algorithm, you may specify the KMS customer master key (CMK) through this configuration. If left unspecified, COS will utilize the default CMK it creates. For further details, please refer to <a href="#">SSE-KMS Encryption</a>.</li> </ul>	-	Not required
	When enabling COS's		

<code>fs.cosn.server-side-encryption.context</code>	SSE-KMS server-side encryption algorithm, you may specify the encryption context through this configuration item, with the value being a key-value pair in JSON format. For example: <code>{"key1":"value1"}</code>	-	Not required
<code>fs.cosn.client-side-encryption.enabled</code>	Whether to enable client-side encryption, which is not enabled by default. After enabling it, you must configure the public key and private key for it, and you cannot use the APPEND and TRUNCATE APIs.	false	Not required
<code>fs.cosn.client-side-encryption.public.key.path</code>	The absolute path of the public key file for client-side encryption.	-	Not required
<code>fs.cosn.client-side-encryption.private.key.path</code>	The absolute path of the private key file for client-side encryption.	-	Not required
<code>fs.cosn.crc64.checksum.enabled</code>	Whether to enable CRC-64 checksum. It is disabled by default, meaning that you can't run the <code>hadoop fs -checksum</code> command to obtain the CRC-64 checksum of a file.	false	Not required
<code>fs.cosn.crc32c.checksum.enabled</code>	Whether to enable CRC32C checksum. It is disabled by default, meaning that you cannot run the <code>hadoop fs -checksum</code> command to obtain the CRC32C checksum of a file. CRC-64 and CRC32C cannot be both enabled.	false	Not required
	The limit on the unload		

<code>fs.cosn.traffic.limit</code>	The limit on the upload bandwidth in bits/s. Value range: 819200-838860800. Default value: <code>-1</code> (unlimited).	-	Not required

## Independent Bucket Configuration Options

**Background:** To support the need for accessing multiple buckets across different regions with varying configurations, a one-time configuration is provided.

**Core principle:** When accessing a bucket, the system prioritizes using its individual configuration settings. If no individual settings are available, it falls back to the original configuration, and if that is not set, it uses the default configuration.

The following example demonstrates how to set the `fs.cosn.upload.buffer` configuration item specifically for the "examplebucket-1250000000" bucket:

```
fs.cosn.bucket.examplebucket-1250000000.upload.buffer ***
```

### Note:

Here, the subkey for `fs.cosn.upload.buffer` (trimming `fs.cosn.`) is `upload.buffer`, so the independent configuration item for the bucket is `fs.cosn.bucket.<bucket-appid>.<subkey>`.

### How to Use:

```
hadoop fs -ls cosn://examplebucket-1250000000/
```

## Simplified Bucket Configuration (Applicable to Standard Buckets)

Access via `cosn://<bucket>/` method.

The following example demonstrates how to set the `fs.cosn.upload.buffer` configuration item specifically for the "examplebucket-1250000000" bucket:

```
fs.cosn.userinfo.appid 1250000000
fs.cosn.bucket.examplebucket.upload.buffer ***
```

### Note:

Here, the subkey for `fs.cosn.upload.buffer` (trimming `fs.cosn.`) is `upload.buffer`. However, since the `appid` has already been set, the independent configuration item for

the bucket can be simplified to `fs.cosn.bucket.<bucket>.<subkey>` .

#### How to Use:

```
hadoop fs -ls cosn://examplebucket/
```

### Simplified Bucket Configuration Options (Applicable to Metadata Acceleration Storage Buckets)

Access via `cosn://<bucket>/` method.

The following example demonstrates how to set the `fs.cosn.upload.buffer` configuration item specifically for the "examplebucket-1250000000" bucket:

```
fs.cosn.userinfo.appid 1250000000
fs.cosn.trsf.fs ofs.use.short.bucketname true
fs.cosn.bucket.examplebucket.upload.buffer ***
```

#### Note:

Here, the subkey for `fs.cosn.upload.buffer` (trimming `fs.cosn.`) is `upload.buffer`. However, since the `appid` has already been set, the independent configuration item for the bucket can be simplified to `fs.cosn.bucket.<bucket>.<subkey>` .

Due to access restrictions on mount point formats in older versions of the OFS Java SDK, only the `<bucket-appid>` method is supported. Therefore, it is necessary to update the dependent plugin to the latest version for use.

Dependency plugin version information:

```
ofs java sdk >= 1.1.8
hadoop cos >= 8.3.0
```

#### How to Use:

```
hadoop fs -ls cosn://examplebucket/
```

## Server-Side Encryption

Hadoop-COS supports server-side encryption. It currently provides three encryption methods: COS-managed key (SSE-COS), KMS managed key (SSE-KMS), and custom key

(SSE-C). The encryption feature in Hadoop-COS is disabled by default. Users can enable it and configure in the following ways.

## SSE-COS encryption

SSE-COS encryption refers to server-side encryption with a COS-managed key. In this mode, COS manages the master key and its data. When using Hadoop-COS, you can add the following configuration in the `$HADOOP_HOME/etc/hadoop/core-site.xml` file to implement SSE-COS encryption.

```
<property>
  <name>fs.cosn.server-side-encryption.algorithm</name>
  <value>SSE-COS</value>
  <description>The server side encryption algorithm.</description>
</property>
```

## SSE-KMS Encryption

SSE-KMS encryption uses KMS-managed keys for server-side encryption. KMS is a Security management service introduced by Tencent Cloud, which uses third-party-certified Hardware Security modules (HSM) to generate and protect keys. It enables users to easily create and manage keys, meeting the key management needs across multiple applications and businesses as well as satisfying regulatory and compliance requirements. When using Hadoop-COS, users can add the following configuration in the `$Hadoop_HOME/etc/Hadoop/core-site.xml` file to implement SSE-KMS encryption.

```
<property>
  <name>fs.cosn.server-side-encryption.algorithm</name>
  <value>SSE-KMS</value>
  <description>The server side encryption algorithm.</description>
</property>
<property>
  <name>fs.cosn.server-side-encryption.key</name>
  <value>23e80852-1e38-11e9-b129-5cb9019b4b01</value>
  <description>The CMK for KMS encryption. This configuration is optional</description>
</property>
<property>
  <name>fs.cosn.server-side-encryption.context</name>
  <value>{"key1": "value1"}</value>
```

```
<description>The KMS encryption context, it's a json string with
key/value pairs. This configuration is optional</description>
</property>
```

## SSE-C encryption

SSE-C encryption refers to server-side encryption with customer-provided keys. That is, the encryption key is provided by the user. When you upload an object, COS will use the encryption key that you provide to apply AES-256 encryption to the data. When using Hadoop-COS, you can add the following configuration in the `$ HADOOP_HOME/etc/hadoop/core-site.xml` file to implement SSE-C encryption.

```
<property>
  <name>fs.cosn.server-side-encryption.algorithm</name>
  <value>SSE-C</value>
  <description>The server side encryption algorithm.</description>
</property>
<property>
  <name>fs.cosn.server-side-encryption.key</name>
  <value>MDEyMzQ1Njc4OUFCQ0RFRjAxMjMONTY3ODlBQkNERUY=</value> #
Users need to configure the SSE-C key themselves, which is a Base64-
encoded AES-256 key.
  <description>The SSE-C server side encryption key.</description>
</property>
```

### Note:

- The SSE-C encryption feature of Hadoop-COS relies on the SSE-C server-side encryption of COS. This means Hadoop-COS does not store any user-defined encryption keys just like COS. Instead, COS stores HMAC values with random data added to the encryption keys to authenticate access requests. COS cannot use the HMAC values to derive the value of an encryption key or decrypt the content of an object. Therefore, if you lose your encryption key, you will not be able to access the object again.
- If you configure an SSE-C server-side encryption algorithm on Hadoop-COS, you must also configure an SSE-C key using the `fs.cosn.server-side-encryption.key` parameter in the format of a Base64-encoded AES-256 key.
- Hadoop-COS reports the exception "Set SSE\_COS request no such method" for non-write requests in v5.10.0.

## Client Encryption

COSN client-side encryption adopts the RSA encryption method. The key is divided into public key and private key. The former is used for file encryption, and the latter is used for file decryption. When a file is uploaded, COSN will generate a random key and use it to encrypt the file symmetrically. The public key encrypts this key and saves the encrypted information in the file's metadata. When the file is downloaded, COSN will use the private key to obtain the encrypted random key from the file's metadata for decryption and then use the decrypted random key to decrypt the file. The public and private keys only participate in the local calculation in the client but are not transferred on the network or stored on the server, ensuring the data security of the master key.

- When using client-side encryption, you should ensure the integrity and accuracy of the master key. When copying or migrating encrypted data, you should ensure the integrity and accuracy of the cryptographic metadata. If any encrypted data cannot be decoded due to cryptographic metadata loss/corruption caused by your incorrect use or loss of the master key, you shall bear all losses and consequences arising from it.
- After client-side encryption is enabled, APPEND and TRUNCATE APIs are no longer supported.
- If you run the `hadoop fs -cp` command on an encrypted file in a client with the client-side encryption feature disabled, the encrypted information will be lost.
- After client-side encryption is enabled, CRC file verification and async multipart upload are disabled by default.

When using Hadoop-COS, you can add the following configuration in the `$HADOOP_HOME/etc/hadoop/core-site.xml` file to implement SSE-COS encryption.

```
<property>
  <name>fs.cosn.client-side-encryption.enabled</name>
  <value>>true</value>
  <description>Enable or disable the client encryption
function</description>
</property>

<property>
  <name>fs.cosn.client-side-encryption.public.key.path</name>
  <value>/xxx/xxx.key</value>
  <description>The direct path to the public key</description>
</property>

<property>
```

```
<name>fs.cosn.client-side-encryption.private.key.path</name>
<value>/xxx/xxx.key</value>
<description>The direct path to the private key</description>
</property>
```

You can generate the key with the following code:

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;

// Use asymmetric key RSA encryption for randomly generated symmetric
keys
public class BuildKey {
    private static final SecureRandom srand = new SecureRandom();
    private static void buildAndSaveAsymKeyPair(String pubKeyPath,
String priKeyPath) throws IOException, NoSuchAlgorithmException {
        KeyPairGenerator keyGenerator =
KeyPairGenerator.getInstance("RSA");
        keyGenerator.initialize(1024, srand);
        KeyPair keyPair = keyGenerator.generateKeyPair();
        PrivateKey privateKey = keyPair.getPrivate();
        PublicKey publicKey = keyPair.getPublic();

        X509EncodedKeySpec x509EncodedKeySpec = new
X509EncodedKeySpec(publicKey.getEncoded());
        FileOutputStream fos = new FileOutputStream(pubKeyPath);
        fos.write(x509EncodedKeySpec.getEncoded());
        fos.close();

        PKCS8EncodedKeySpec pkcs8EncodedKeySpec = new
PKCS8EncodedKeySpec(privateKey.getEncoded());
```

```
fos = new FileOutputStream(priKeyPath);
fos.write(pkcs8EncodedKeySpec.getEncoded());
fos.close();
}

public static void main(String[] args) throws Exception {

    String pubKeyPath = "pub.key";
    String priKeyPath = "pri.key";
    buildAndSaveAsymKeyPair(pubKeyPath, priKeyPath);
}
}
```

## Setting Classpath for `Yarn-Site.Xml`

```
yarn-site.xml add
```

```
<property>
  <name>yarn.application.classpath</name>
  <value>$SHADOOP_CLASSPATH</value>
</property>
```

## How to Use

### Use Cases

The command format is `hadoop fs -ls -R cosn://<BucketName-APPID>/<path>` or `hadoop fs -ls -R /<path>` (requires configuring the `fs.defaultFS` option as `cosn://BucketName-APPID`). In the following example, a bucket named `examplebucket-1250000000` is used, and a specific path can be added after it.

```
hadoop fs -ls -R cosn://examplebucket-1250000000/
-rw-rw-rw-  1 root root      1087 2018-06-11 07:49
cosn://examplebucket-1250000000/LICENSE
drwxrwxrwx  - root root          0 1970-01-01 00:00
cosn://examplebucket-1250000000/hdfs
drwxrwxrwx  - root root          0 1970-01-01 00:00
cosn://examplebucket-1250000000/hdfs/2018
```

```
-rw-rw-rw-  1 root root      1087 2018-06-12 03:26
cosn://examplebucket-1250000000/hdfs/2018/LICENSE
-rw-rw-rw-  1 root root      2386 2018-06-12 03:26
cosn://examplebucket-1250000000/hdfs/2018/ReadMe
drwxrwxrwx  - root root          0 1970-01-01 00:00
cosn://examplebucket-1250000000/hdfs/test
-rw-rw-rw-  1 root root      1087 2018-06-11 07:32
cosn://examplebucket-1250000000/hdfs/test/LICENSE
-rw-rw-rw-  1 root root      2386 2018-06-11 07:29
cosn://examplebucket-1250000000/hdfs/test/ReadMe
```

Run the wordcount provided in MapReduce, and execute the following command.

**Note:**

In the following commands, `hadoop-mapreduce-examples-2.7.2.jar` is used as an example for version 2.7.2. If you are using a different version, please replace it with the corresponding version number.

```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-
2.7.2.jar wordcount cosn://example/mr/input cosn://example/mr/output3
```

It will return the following statistics if it executes successfully:

```
File System Counters
COSN: Number of bytes read=72
COSN: Number of bytes written=40
COSN: Number of read operations=0
COSN: Number of large read operations=0
COSN: Number of write operations=0
FILE: Number of bytes read=547350
FILE: Number of bytes written=1155616
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=0
HDFS: Number of bytes written=0
HDFS: Number of read operations=0
HDFS: Number of large read operations=0
```

```
HDFS: Number of write operations=0
Map-Reduce Framework
Map input records=5
Map output records=7
Map output bytes=59
Map output materialized bytes=70
Input split bytes=99
Combine input records=7
Combine output records=6
Reduce input groups=6
Reduce shuffle bytes=70
Reduce input records=6
Reduce output records=6
Spilled Records=12
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=0
Total committed heap usage (bytes)=653262848
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=36
File Output Format Counters
Bytes Written=40
```

## Accessing COSN through Java code

```
package com.qcloud.chdfs.demo;

import org.apache.commons.io.IOUtils;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FSDataOutputStream;
```

```
import org.apache.hadoop.fs.FileChecksum;
import org.apache.hadoop.fs.FileStatus;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;

import java.io.IOException;
import java.net.URI;
import java.nio.ByteBuffer;

public class Demo {
    private static FileSystem initFS() throws IOException {
        Configuration conf = new Configuration();
        // For COSN configuration options, please refer to
https://cloud.tencent.com/document/product/436/6884#hadoop-.E9.85.8D.E7.
BD.AE
        // The following configurations are required
        conf.set("fs.cosn.impl", "org.apache.hadoop.fs.CosFileSystem");
        conf.set("fs.AbstractFileSystem.cosn.impl",
"org.apache.hadoop.fs.CosN");
        conf.set("fs.cosn.tmp.dir", "/tmp/hadoop_cos");
        conf.set("fs.cosn.bucket.region", "ap-guangzhou");
        conf.set("fs.cosn.userinfo.secretId", "AKXXXXXXXXXXXXXXXXXXXX");
        conf.set("fs.cosn.userinfo.secretKey", "XXXXXXXXXXXXXXXXXXXX");
        conf.set("fs ofs.user.appid", "XXXXXXXXXXXX");
        // For additional configuration references, please visit the
official documentation:
https://cloud.tencent.com/document/product/436/6884#hadoop-.E9.85.8D.E7.
BD.AE
        // Enable CRC64 checksum. By default, it is disabled, and the
CRC64 checksum of a file cannot be obtained using the 'hadoop fs -
checksum' command.
        conf.set("fs.cosn.crc64.checksum.enabled", "true");
        String cosnUrl = "cosn://f4mxxxxxxxx-125xxxxxxxx";
        return FileSystem.get(URI.create(cosnUrl), conf);
    }

    private static void mkdir(FileSystem fs, Path filePath) throws
IOException {
        fs.mkdirs(filePath);
    }
}
```

```
private static void createFile(FileSystem fs, Path filePath) throws
IOException {
    // Create a new file (overwrite if it already exists)
    // if the parent dir does not exist, fs will create it!
    FSDataOutputStream out = fs.create(filePath, true);
    try {
        // Writing a file
        String content = "test write file";
        out.write(content.getBytes());
    } finally {
        IOUtils.closeQuietly(out);
    }
}

private static void readFile(FileSystem fs, Path filePath) throws
IOException {
    FSDataInputStream in = fs.open(filePath);
    try {
        byte[] buf = new byte[4096];
        int readLen = -1;
        do {
            readLen = in.read(buf);
        } while (readLen >= 0);
    } finally {
        IOUtils.closeQuietly(in);
    }
}

private static void queryFileOrDirStatus(FileSystem fs, Path path)
throws IOException {
    FileStatus fileStatus = fs.getFileStatus(path);
    if (fileStatus.isDirectory()) {
        System.out.printf("path %s is dir\n", path);
        return;
    }

    long fileLen = fileStatus.getLen();
    long accessTime = fileStatus.getAccessTime();
    long modifyTime = fileStatus.getModificationTime();
    String owner = fileStatus.getOwner();
}
```

```
String group = fileStatus.getGroup();

    System.out.printf("path %s is file, fileLen: %d, accessTime: %d,
modifyTime: %d, owner: %s, group: %s\n",
        path, fileLen, accessTime, modifyTime, owner, group);
}

private static void getFileChecksum(FileSystem fs, Path path) throws
IOException {
    FileChecksum checksum = fs.getFileChecksum(path);
    System.out.printf("path %s, checkSumType: %s, checkSumCrcVal:
%d\n",
        path, checksum.getAlgorithmName(),
        ByteBuffer.wrap(checksum.getBytes()).getInt());
}

private static void copyFileFromLocal(FileSystem fs, Path cosnPath,
Path localPath) throws IOException {
    fs.copyFromLocalFile(localPath, cosnPath);
}

private static void copyFileToLocal(FileSystem fs, Path cosnPath,
Path localPath) throws IOException {
    fs.copyToLocalFile(cosnPath, localPath);
}

private static void renamePath(FileSystem fs, Path oldPath, Path
newPath) throws IOException {
    fs.rename(oldPath, newPath);
}

private static void listDirPath(FileSystem fs, Path dirPath) throws
IOException {
    FileStatus[] dirMemberArray = fs.listStatus(dirPath);

    for (FileStatus dirMember : dirMemberArray) {
        System.out.printf("dirMember path %s, fileLen: %d\n",
dirMember.getPath(), dirMember.getLen());
    }
}
```

```
// Recursive flag for deleting directories
If recursive is set to false and the directory is not empty, the
operation will fail.
private static void deleteFileOrDir(FileSystem fs, Path path,
boolean recursive) throws IOException {
    fs.delete(path, recursive);
}

private static void closeFileSystem(FileSystem fs) throws
IOException {
    fs.close();
}

public static void main(String[] args) throws IOException {
    // Initialize the file
    FileSystem fs = initFS();

    // Create a file
    Path cosnFilePath = new Path("/folder/exampleobject.txt");
    createFile(fs, cosnFilePath);

    // Reading a file
    readFile(fs, cosnFilePath);

    // Query file or directory
    queryFileOrDirStatus(fs, cosnFilePath);

    // Obtain the file checksum
    getFileChecksum(fs, cosnFilePath);

    // Copy files from local storage
    Path localFilePath = new
Path("file:///home/hadoop/ofs_demo/data/exampleobject.txt");
    copyFileFromLocal(fs, cosnFilePath, localFilePath);

    // Retrieve the file to local
    Path localDownFilePath = new
Path("file:///home/hadoop/ofs_demo/data/exampleobject.txt");
    copyFileToLocal(fs, cosnFilePath, localDownFilePath);
}
```

```
listDirPath(fs, cosnFilePath);
// Renaming
mkdir(fs, new Path("/doc"));
Path newPath = new Path("/doc/example.txt");
renamePath(fs, cosnFilePath, newPath);

// Delete a file
deleteFileOrDir(fs, newPath, false);

// Create directory
Path dirPath = new Path("/folder");
mkdir(fs, dirPath);

// Create a file in the directory
Path subFilePath = new Path("/folder/exampleobject.txt");
createFile(fs, subFilePath);

// List the directories
listDirPath(fs, dirPath);

// Delete directory
deleteFileOrDir(fs, dirPath, true);
deleteFileOrDir(fs, new Path("/doc"), true);

// Close the file system
closeFileSystem(fs);
}
}
```

## FAQs

If you have any questions about Hadoop-COS, see [FAQs > Tools > Hadoop](#).

# COSDistCp

Last updated: 2023-09-07 12:29:30

## Note

COSDistCp is a MapReduce-based distributed file copy tool mainly used for data copy between HDFS and COS. It introduces the following features:

- Performs incremental file migration and data verification based on length and CRC checksum.
- Filters files in the source directory with regular expression.
- Decompresses files in the source directory and compresses them to the target compression format.
- Aggregates text files based on a regular expression.
- Preserves user/user group, extension attributes, and time of the source file and directory.
- Configures alarms and Prometheus monitoring.
- Collects the statistics of file size distribution.
- Limits the read bandwidth.

## Operating Environment

### System environment

Linux.

### Software requirements

Hadoop 2.6.0 or above; Hadoop-COS 5.9.3 or above.

## Download and Installation

### Obtaining the COSDistCp JAR package

- Hadoop 2.x users can download the [cos-distcp-1.13-2.8.5.jar package](#) and verify the integrity of the downloaded jar package using the [MD5 checksum value](#).
- Hadoop 3.x users can download the [cos-distcp-1.13-3.1.0.jar package](#) and verify the integrity of the downloaded JAR file using the [MD5 checksum value](#).

## Notes

In the Hadoop environment, install [Hadoop-COS](#) and then run the COSDistCp tool. For users who have not installed and configured the Hadoop-COS plugin in their environment, download the corresponding COSDistCp jar, Hadoop-COS jar, and cos\_api-bundle jar

packages according to the Hadoop version (see the download links above). Specify the Hadoop-COS related parameters to execute the copy task, where the jar package address should be the local jar location.

### Note

The Hadoop-COS jar and `cos_api-bundle` jar files must be placed in one of the paths specified by the `hadoop classpath` command, such as `/usr/local/services/hadoop/share/hadoop/tools/lib/`.

```
hadoop jar cos-distcp-${version}.jar \  
-libjars /usr/local/services/hadoop/share/hadoop/tools/lib/cos_api-  
bundle-${version}.jar,/usr/local/services/hadoop/share/hadoop/tools/lib/  
hadoop-cos-${version}.jar \  
-  
Dfs.cosn.credentials.provider=org.apache.hadoop.fs.auth.SimpleCredential  
Provider \  
-Dfs.cosn.userinfo.secretId=COS_SECRETID \  
-Dfs.cosn.userinfo.secretKey=COS_SECRETKEY \  
-Dfs.cosn.bucket.region=ap-guangzhou \  
-Dfs.cosn.impl=org.apache.hadoop.fs.CosFileSystem \  
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \  
--src /data/warehouse \  
--dest cosn://examplebucket-1250000000/warehouse
```

## How It Works

COSDistCp uses the MapReduce framework. The multi-process and multi-thread tool performs operations such as file copy, data verification, compression, file attribute preservation, and copy retries. COSDistCp will overwrite files with the same name in the destination location. If data copy or verification fails, the corresponding file may fail to be copied and information about these files will be written in a temporary directory. If new files are added to your source file system or the file content changes, you can use the `--skipMode` or `--diffMode` parameter to compare the length or CRC checksum of the files to implement data verification and incremental file migration.

## Description

As a Hadoop user, you can run the command `hadoop jar cos-distcp-${version}.jar --help` to view the supported parameter options for COSDistCp, where `${version}` represents the

version number. The following are the parameter descriptions for the current version of COSDistCp:

Attribute Key	Note	Default value	Required
<code>--help</code>	Display supported parameter options for COSDistCp. Example: <code>--help</code>	–	Not required
<code>--src=LOCATION</code>	Specifies the source directory for copying, which can be either an HDFS or COS path. Example: <code>--src=hdfs://user/logs/</code>	–	Required
<code>--dest=LOCATION</code>	Specifies the target directory for copying, which can be either an HDFS or COS path. Example: <code>--dest=cosn://examplebucket-1250000000/user/logs</code>	–	Required
<code>--srcPattern=PATTERN</code>	Specify a regular expression to filter files in the source directory. Example: <code>--srcPattern='.*\.log\$'</code> <b>Note: Enclose parameters in single quotes to avoid the * symbol being interpreted by the shell.</b>	–	Not required
<code>--taskNumber=VALUE</code>	Number of copy threads Example: <code>--taskNumber=10</code>	10	Not required
<code>--workerNumber=VALUE</code>	Specifies the number of copy threads, and COSDistCp creates a copy thread pool of this size in each copy process. Example: <code>--workerNumber=4</code>	4	Not required
<code>--filesPerMapper=VALUE</code>	Specifies the number of input rows per Mapper. Example: <code>--filesPerMapper=10,000</code>	500000	Not required
<code>--groupBy=PATTERN</code>	Aggregate text files using specified regular expressions. Example: <code>--groupBy='.*group-input/(\d+)-(\d+).*</code>	–	Not required
<code>--targetSize=VALUE</code>	Specifies the target file size in MB, used in conjunction with <code>--groupBy</code> .	–	Not required

<code>targetSize=VALUE</code>	Example: <code>--targetSize=10</code>		required
<code>--outputCodec=VALUE</code>	<p>Specifies the compression method for output files, with options including <code>gzip</code>, <code>lzo</code>, <code>snappy</code>, <code>none</code>, and <code>keep</code>, where:</p> <ol style="list-style-type: none"> <li>Retains the original compression method of the files.</li> <li>If set to 'none', files will be decompressed based on their file extensions.</li> <li>Example: <code>--outputCodec=gzip</code></li> <li>Note: If there are files <code>/dir/test.gzip</code> and <code>/dir/test.gz</code>, and the output format is specified as <code>lzo</code>, only one file <code>/dir/test.lzo</code> will be retained.</li> </ol>	<code>keep</code>	Not required
<code>--deleteOnSuccess</code>	<p>Immediately deletes the source file upon successful copy to the target directory.</p> <p>Example: <code>--deleteOnSuccess</code>,  <b>Important: Versions 1.7 and above no longer provide this parameter. It is recommended to delete the data in the source file system after successful data migration and verification using <code>--diffMode</code></b></p>	<code>false</code>	Not required
<code>--multipartUploadChunkSize=VALUE</code>	<p>Specifies the block size for the Hadoop-COS plugin to transfer files to COS. The maximum number of blocks supported by COS is 10,000. You can adjust the block size based on the file size, in MB. The default block size is 8 MB.</p> <p>Example: <code>--multipartUploadChunkSize=20</code></p>	<code>8MB</code>	Not required
<code>--cosServerSideEncryption</code>	<p>Specifies SSE-COS as the encryption and decryption algorithm when uploading files to COS.</p> <p>Example: <code>--cosServerSideEncryption</code></p>	<code>false</code>	Not required
	Upon completion of the copy,		

<code>--outputManifest=VALUE</code>	generates a list of target files for the current copy in the destination directory (GZIP compressed). Example: <code>--outputManifest=manifest.gz</code>	-	Not required
<code>--requirePreviousManifest</code>	Requires specifying the <code>--previousManifest=VALUE</code> parameter for incremental copy. Example: <code>--requirePreviousManifest</code>	false	Not required
<code>--previousManifest=LOCATION</code>	Information on the target files generated from the previous copy. Example: <code>--previousManifest=cosn://examplebucket-1250000000/big-data/manifest.gz</code>	-	Not required
<code>--copyFromManifest</code>	When used with <code>--previousManifest=LOCATION</code> , it copies files from the <code>--previousManifest</code> to the target file system. Example: <code>--copyFromManifest</code>	false	Not required
<code>--storageClass=VALUE</code>	Specify the object storage type, with options including STANDARD, STANDARD_IA, ARCHIVE, DEEP_ARCHIVE, and INTELLIGENT_TIERING. For more information on supported storage types, see <a href="#">Storage Class Overview</a> .	-	Not required
<code>--srcPrefixesFile=LOCATION</code>	Specify a local file, where each line contains a source directory to be copied. Example: <code>--srcPrefixesFile=file:///data/migrate-folders.txt</code>	-	Not required
<code>--skipMode=MODE</code>	Before copying, verify whether the source and target files are the same. If they are, skip the copy. Options include none (no verification), length, checksum (CRC value), length-mtime (length + mtime value), and length-checksum (length + CRC	length-checksum	Not required

	value). Example: <code>--skipMode=length</code>		
<code>--checkMode=MODE</code>	Upon file copy completion, verifies whether the source and target files are identical. Options include none (no verification), length, checksum (CRC value), length-mtime (length + mtime value), and length-checksum (length + CRC value). Example: <code>--checkMode=length-checksum</code>	length-checksum	Not required
<code>--diffMode=MODE</code>	Specifies the difference file list between source and destination directories, with options for length, checksum (CRC value), length-mtime (length + mtime value), and length-checksum (length + CRC value). Example: <code>--diffMode=length-checksum</code>	-	Not required
<code>--diffOutput=LOCATION</code>	Specify the HDFS output directory in diffMode, which must be empty. Example: <code>--diffOutput=/diff-output</code>	-	Not required
<code>--cosChecksumType=TYPE</code>	Specifies the CRC algorithm used by the Hadoop-COS plugin, with options including CRC32C and CRC64. Example: <code>--cosChecksumType=CRC32C</code>	CRC32C	Not required
<code>--preserveStatus=VALUE</code>	Specifies whether to copy the source file's user, group, permission, xattr, and timestamps metadata to the target file, with optional values being ugpxt (i.e., the first letters of user, group, permission, xattr, and timestamps). Example: <code>--preserveStatus=ugpt</code>	-	Not required
<code>--ignoreSrcMiss</code>	Ignores files that exist in the manifest file but cannot be found during the copy.	false	Not required

<code>--promGatewayAddress=VALUE</code>	Specifies the Prometheus PushGateway address and port for pushing the counter data of MapReduce jobs.	-	Not required
<code>--promGatewayDeleteOnFinish=VALUE</code>	Upon task completion, delete the metric collection with JobName in the Prometheus PushGateway. Example: <code>--promGatewayDeleteOnFinish=true</code>	true	Not required
<code>--promGatewayJobName=VALUE</code>	Specifies the JobName to report to Prometheus PushGateway. Example: <code>--promGatewayJobName=cos-distcp-hive-backup</code>	-	Not required
<code>--promCollectInterval=VALUE</code>	Specifies the interval for collecting MapReduce task Counter information, in milliseconds (ms). Example: <code>--promCollectInterval=5000</code>	5000	Not required
<code>--promPort=VALUE</code>	Designate the server port for exposing Prometheus metrics externally. Example: <code>--promPort=9028</code>	-	Not required
<code>--enableDynamicStrategy</code>	Enables dynamic task allocation strategy, allowing faster migration tasks to process more files for increased efficiency. <b>Note: This mode has certain limitations, such as inaccurate task counters in case of process anomalies. Please verify the data using <code>--diffMode</code> after the migration is complete.</b> Example: <code>--enableDynamicStrategy</code>	false	Not required
<code>--splitRatio=VALUE</code>	Specifies the split ratio for the Dynamic Strategy; the larger the splitRatio value, the finer the task granularity. Example: <code>--splitRatio=8</code>	8	Not required
<code>--</code>	Specifies the local folder where the task information files generated by		Not

<code>localTemp=VALUE</code>	task information files generated by Dynamic Strategy are located. Example: <code>--localTemp=/tmp</code>	<code>/tmp</code>	Not required
<code>--taskFilesCopyThreadNum=VALUE</code>	Specifies the concurrency of Dynamic Strategy task information file copy to HDFS. Example: <code>--taskFilesCopyThreadNum=32</code>	32	Not required
<code>--statsRange=VALUE</code>	Specify the range for statistics. Example: <code>---statsRange=0,1MB,10MB,100MB,1GB,10GB,inf</code>	0,1mb,10mb,100mb,1gb,10gb,inf	Not required
<code>--printStatsOnly</code>	Gathers distribution information of files to be migrated without actually migrating the data. Example: <code>--printStatsOnly</code>	-	Not required
<code>--bandWidth</code>	Limits the bandwidth for reading each migrated file, measured in MB/s; default is -1, indicating no bandwidth restriction. Example: <code>--bandWidth=10</code>	-	Not required
<code>--jobName</code>	Specify the name of the migration task. Example: <code>--jobName=cospdistcp-to-warehouse</code>	-	Not required
<code>--compareWithCompatibleSuffix</code>	When using <code>--skipMode</code> and <code>--diffMode</code> parameters, determine whether to convert the source file's gzip extension to gz, and lzop file extension to lzo. Example: <code>--compareWithCompatibleSuffix</code>	-	Not required
<code>--delete</code>	Ensures consistency between source and destination directories by moving files present in the destination but not in the source to a separate trash directory, while generating a file inventory. Note: The <code>--diffMode</code> parameter cannot be used simultaneously.	-	Not required

<code>--deleteOutput</code>	Specify the HDFS output directory for deletion, which must be empty. Example: <code>--deleteOutput=/dele-output</code>	-	Not required
-----------------------------	---	---	--------------

## Use Cases

### Viewing the help option

Run the following command with `--help` to view the parameters supported by COSDistCp:

```
hadoop jar cos-distcp-${version}.jar --help
```

In the above command, `${version}` represents the COSDistCp version number, for example, the COSDistCp jar package name for version 1.0 is `cos-distcp-1.0.jar`.

### File size distribution of the files to copy

Run the following command with the `--printStatsOnly` and `--statsRange=VALUE` parameters to output the file size distribution of the files to copy:

```
hadoop jar cos-distcp-${version}.jar --src /wookie/data --dest
cosn://examplebucket-1250000000/wookie/data --printStatsOnly --
statsRange=0,1mb,10mb,100mb,1gb,10gb,inf
```

Copy File Distribution Statistics:

Total File Count: 4

Total File Size: 1190133760

SizeRange	TotalCount	TotalSize
0MB ~ 1MB	0 (0.00%)	0 (0.00%)
1MB ~ 10MB	1 (25.00%)	1048576 (0.09%)
10MB ~ 100MB	1 (25.00%)	10485760 (0.88%)
100MB ~ 1024MB	1 (25.00%)	104857600 (8.81%)
1024MB ~ 10240MB	1 (25.00%)	1073741824 (90.22%)
10240MB ~ LONG_MAX	0 (0.00%)	0 (0.00%)

### Specifying the source and destination locations for the files to copy

Run the following command with the `--src` and `--dest` parameters:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
```

```
cosn://examplebucket-1250000000/data/warehouse
```

COSDistCp will retry 5 times for files that failed to be copied. If the copy still fails, these files will be written to the `/tmp/${randomUUID}/output/failed/` directory, where `${randomUUID}` is a random string. After recording the failed file information, COSDistCp will continue to migrate the remaining files, and the migration task will not fail due to the migration failure of some files. When the migration task is completed, COSDistCp will output counter information (ensure that your task submitting machine is configured with INFO log output for MapReduce jobs on the submission end) and determine whether there are files that failed to be migrated, and if yes, it will throw an exception on the client that submitted the task.

The following information about a source file might be contained in the output:

1. SRC\_MISS: The copy fails because the source file contained in the manifest is not found.
2. COPY\_FAILED: The copy fails due to other reasons.

You can run the copy command again to implement incremental migration. Run the following command to obtain the log of the MapReduce job. In this way, you can find out the cause of the copy failure. Note that `application_1610615435237_0021` is the application ID.

```
yarn logs -applicationId application_1610615435237_0021 >  
application_1610615435237_0021.log
```

## Querying counters

When the copy operation ends, statistics on the copy will be output. The counters are as follows:

```
CosDistCp Counters  
  BYTES_EXPECTED=10198247  
  BYTES_SKIPPED=10196880  
  FILES_COPIED=1  
  FILES_EXPECTED=7  
  FILES_FAILED=1  
  FILES_SKIPPED=5
```

The statistics are described as follows:

Statistics Item	Note
BYTES_EXPECTED	Total size (in bytes) to copy according to the source directory
FILES_EXPECTED	Number of files to copy according to the source directory, including the directory itself
BYTES_SKIPPED	Total size (in bytes) of files that can be skipped (same length or checksum value)
FILES_SKIPPED	Number of source files that can be skipped (same length or checksum value)
FILES_COPIED	Number of source files that are successfully copied
FILES_FAILED	Number of source files that failed to be copied
FOLDERS_COPIED	Number of directories that are successfully copied
FOLDERS_SKIPPED	Number of directories that are skipped

## Specifying the number of copy processes and the number of threads in each process

Run the following command with the `--taskNumber` and `--workersNumber` parameters. COSDistCp adopts a multi-process, multi-thread framework for the copy operation. You can:

- Use `--taskNumber` to specify the number of processes.
- Use `--workerNumber` to specify the number of threads in each copy process.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest
cosn://examplebucket-1250000000/data/warehouse --taskNumber=10 --
workerNumber=5
```

## Skipping files with the same check value for incremental migration

Execute the command with the parameter `--skipMode` to skip copying files with the same length and checksum between the source and target. The default value is length-checksum:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --skipMode=length-
checksum
```

`--skipMode` option is used to verify if the source and target files are identical before copying. If they are, the copy is skipped. Available options are none (no verification), length, checksum (CRC value), and length-checksum (length + CRC value).

If the checksum algorithms of the source and destination file systems are different, the source file will be read for calculating a new checksum. If your source is HDFS, you can identify whether the HDFS source supports the COMPOSITE-CRC32C algorithm as follows:

```
hadoop fs -Ddfs.checksum.combine.mode=COMPOSITE_CRC -checksum
/data/test.txt
/data/test.txt COMPOSITE-CRC32C 6a732798
```

## Verifying data after migration and migrating incremental data

Run the command with the `--diffMode` and `--diffOutput` parameters:

- `--diffMode` offers two optional values: length and length-checksum.
  - `--diffMode=length` obtains the list of different files based on whether the file sizes are the same.
  - `--diffMode=length-checksum` obtains the list of different files based on whether the file size and CRC checksum are the same.
- `--diffOutput` specifies the output directory for the diff operation.

If the target file system is COS and the source file system's CRC algorithm is different, COSDistCp will fetch the source file and calculate the target file system's CRC for comparison using the same CRC algorithm value. In the following example, after the migration is completed, the `--diffMode` parameter is used to verify whether the source and target files are the same based on file size and CRC value:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --diffMode=length-
checksum --diffOutput=/tmp/diff-output
```

After the above command is executed successfully, the counter information based on the file list of the source file system will be output (ensure that your task submitting machine is configured with INFO log output for MapReduce jobs on the submission end). You can analyze whether the source and destination files are the same based on the counter information as detailed below:

1. SUCCESS: the source and destination files are the same.
2. DEST\_MISS: The destination file does not exist.
3. SRC\_MISS: The source file contained in the source file manifest is not found during the verification.

4. LENGTH\_DIFF: Sizes of the source and destination files are different.
5. CHECKSUM\_DIFF: CRC checksums of the source and destination files are different.
6. DIFF\_FAILED: The `diff` operation fails due to insufficient permissions or other reasons.
7. TYPE\_DIFF: the source is a directory but the destination is a file.

Additionally, COSDistcp generates a differential file list in the HDFS directory `/tmp/diff-output/failed` (for versions 1.0.5 and earlier, it is `/tmp/diff-output`). You can obtain the differential file list, excluding SRC\_MISS, using the following command:

```
hadoop fs -getmerge /tmp/diff-output/failed diff-manifest
grep -v "comment":"SRC_MISS" diff-manifest |gzip > diff-manifest.gz
```

Run the following command to implement incremental copy based on the list of different files:

```
hadoop jar cos-distcp-${version}.jar --taskNumber=20 --src
/data/warehouse --dest cosn://examplebucket-1250000000/data/warehouse/ -
-previousManifest=file:///usr/local/service/hadoop/diff-manifest.gz --
copyFromManifest
```

After incremental migration is completed, run the command with the `--diffMode` parameter again to check whether the files are completely identical.

## Verifying whether the source and destination files have the same CRC checksum

Execute the command with parameter `--checkMode`. Upon file copy completion, verify if the source and target file lengths and checksums are consistent. The default value is `length-checksum`.

When you are copying files from a non-COS file system to COS, if the CRC algorithms of the source and Hadoop-COS are different, the CRC checksum will be calculated during the copy operation. When the copy operation is completed, the CRC checksum of the destination file will be obtained and compared with the calculated CRC checksum of the source file.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --checkMode=length-
checksum
```

### Note

Effective when `--groupBy` is not specified and `--outputCodec` is set to the default value.

## Restricting the read bandwidth for a single file

Run the command with the `--bandWidth` parameter (in MB). The following example command restricts the read bandwidth of each copied file to 10 MB/s:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --bandWidth=10
```

## Copying multiple directories

You can create a local file (for example, `srcPrefixes.txt`) and add the absolute paths of multiple directories to copy to the file (the directories cannot be in parent-child relationships). After this, you can run the `cat` command to view the directories as follows:

```
cat srcPrefixes.txt
/data/warehouse/20181121/
/data/warehouse/20181122/
```

Then, you can use `--srcPrefixesFile` to specify this file. The command is as follows:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --
srcPrefixesFile file:///usr/local/service/hadoop/srcPrefixes.txt --dest
cosn://examplebucket-1250000000/data/warehouse/ --taskNumber=20
```

## Filtering source files with a regular expression

Run the following command with the `--srcPattern` parameter. In this example, only files whose extension is ".log" in the `/data/warehouse/` directory are copied.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest
cosn://examplebucket-1250000000/data/warehouse --srcPattern='.*\.log$'
```

Do not copy files whose extension is ".temp" or ".tmp":

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest
cosn://examplebucket-1250000000/data/warehouse/ --srcPattern='.*(?
```

```
<!\.temp|\.tmp) $'
```

## Specifying the checksum type of Hadoop-COS

Execute the command with the parameter `--cosChecksumType` , defaulting to `CRC32C`, with options of `CRC32C` and `CRC64`.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest  
cosn://examplebucket-1250000000/data/warehouse --cosChecksumType=CRC32C
```

## Specifying the storage class for COS objects

Run the following command with the `--storageClass` parameter:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest  
cosn://examplebucket-1250000000/data/warehouse/ --  
outputManifest=manifest-2020-01-10.gz --storageClass=STANDARD_IA
```

## Specifying the output compression codec

Execute the command with the parameter `--outputCodec` to compress data in HDFS and back it up to COS in real-time, saving storage costs. The available options for this parameter are: `keep`, `none`, `gzip`, `lzop`, and `snappy`. The 'none' option saves the target file in an uncompressed state, while 'keep' retains the original file's compression status. For example:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/logs --dest  
cosn://examplebucket-1250000000/data/warehouse/logs-gzip --  
outputCodec=gzip
```

### Note

Except for the "keep" option, files are first decompressed and then converted to the target compression type. As a result, the destination file may differ from the source file due to varying compression parameters, but the decompressed files will be consistent. When the `--groupBy` option is not specified and the `--outputCodec` is set to the default value, incremental migration can be performed using `--skipMode`, and data verification can be conducted using `--checkMode`.

## Deleting the source files

Run the command with the `--deleteOnSuccess` parameter. The following example deletes the corresponding source files in the `/data/warehouse` directory immediately after they are copied from HDFS to COS:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --deleteOnSuccess
```

#### Note

When this option is specified, the corresponding source file is deleted immediately after each file is migrated, rather than waiting for the entire migration to complete. Use this option with caution. This parameter is no longer available in version 1.7 and later.

## Generating the target manifest and specifying the previous manifest

Run the command with the `--outputManifest` and `--previousManifest` parameters.

- `--outputManifest` This option first generates a gzip-compressed manifest.gz locally and moves it to the directory specified by `--dest` upon successful migration.
- `--previousManifest` specifies the destination files that are copied during the previous copy operation ( `--outputManifest` ). COSDistCp will skip files of the same size.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --
outputManifest=manifest.gz --previousManifest= cosn://examplebucket-
1250000000/data/warehouse/manifest-2020-01-10.gz
```

#### Note

The incremental migration in the above command can only synchronize files with size changes and cannot synchronize files with content changes. If the file content may change, please refer to the `--diffMode` usage example to determine the list of changed files based on their CRC.

## Using dynamic strategy for migration job distribution

If your files differ greatly in size, (e.g., there are a few large files, causing imbalanced loads of a large number of small files and machines), you can use `--enableDynamicStrategy` to enable the dynamic strategy, which allows faster-speed jobs to copy more files to speed up the whole copy process.

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --enableDynamicStrategy
```

Verify the migrated data after migration is completed:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --diffMode=length-
checksum --diffOutput=/tmp/diff-output
```

### Note

This mode has certain limitations, such as potentially inaccurate task counters in case of process anomalies. Please verify the data using `--diffMode` after the migration is complete.

## Copying metadata of the source file

Execute the command with the `--preserveStatus` parameter to copy the user, group, permission, and timestamps (modification time and access time) of the source file or source directory to the target file or target directory. This parameter is effective when copying files from HDFS to CHDFS.

Example:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --preserveStatus=ugpt
```

## Configuring Prometheus

You can go to YARN to view the COSDistCp job counter, including the number of files/bytes that have been copied. To easily view the graph of the COSDistCp jobs, you can display the data using Prometheus and Grafana with easy configurations. The following example configures `prometheus.yml` to add the jobs to grab:

```
- job_name: 'cos-distcp-hive-backup'
  static_configs:
    - targets: ['172.16.16.139:9028']
```

Run the command with the `--promPort=VALUE` parameter to expose the counter of the current MapReduce job:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --promPort=9028
```

Download the sample [Grafana Dashboard](#) and import it. The Grafana display is as follows:



## Alarms for copy failures

Run the command with the `--completionCallbackClass` parameter to specify the path of the callback class. When the task is completed, COSDistCp will use the collected task information as parameters to execute the callback function. For user-defined callback functions, the following APIs need to be implemented. You can download the [callback sample code](#).

```
package com.qcloud.cos.distcp;
import java.util.Map;
public interface TaskCompletionCallback {
/**
 * @description: When the task is completed, the callback function is
executed
 * @param jobType Copy or Diff
 * @param jobStartTime the job start time
 * @param errorMsg the exception error msg
 * @param applicationId the MapReduce application id
 * @param: cosDistCpCounters the job
 */
void doTaskCompletionCallback(String jobType, long jobStartTime, String
errorMsg, String applicationId, Map<String, Long> cosDistCpCounters);
/**
 * @description: init callback config before execute
 */
}
```

```
void init() throws Exception;
}
```

COSDistCp integrates Tencent Cloud's TCOP alarm system to trigger alerts when anomalies occur or file copy failures are detected during the task execution:

```
export alarmSecretId=SECRET-ID
export alarmSecretKey=SECRET-KEY
export alarmRegion=ap-guangzhou
export alarmModule=module
export alarmPolicyId=cm-xxx
hadoop jar cos-distcp-1.4-2.8.5.jar \
-
Dfs.cosn.credentials.provider=org.apache.hadoop.fs.auth.SimpleCredential
Provider \
-Dfs.cosn.userinfo.secretId=SECRET-ID \
-Dfs.cosn.userinfo.secretKey=SECRET-KEY \
-Dfs.cosn.bucket.region=ap-guangzhou \
-Dfs.cosn.impl=org.apache.hadoop.fs.CosFileSystem \
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \
--src /data/warehouse \
--dest cosn://examplebucket-1250000000/data/warehouse/ \
--checkMode=checksum \
--
completionCallbackClass=com.qqcloud.cos.distcp.DefaultTaskCompletionCallb
ack
```

In the above command, `alarmPolicyId` refers to the TCOP alarm policy. You can create and configure it in the TCOP console (Alarm Management > Alarm Configuration > Custom Message).

## FAQs

### What stages are involved in migration of HDFS data with COSDistCp? How do I adjust the migration performance and ensure the data correctness?

COSDistCp verifies each migrated file upon migration completion according to `checkMode` :

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --taskNumber=20
```

After migration is completed, you can also run the following command to view the list of different source and destination files:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --diffMode=length-
checksum --diffOutput=/tmp/diff-output
```

## How can I run COSDistCp if Hadoop-COS is not configured in the environment?

You can download a specific version of the COSDistCp JAR package according to the Hadoop version and specify the Hadoop-COS-related parameters to perform the copy operation.

```
hadoop jar cos-distcp-${version}.jar \
-
Dfs.cosn.credentials.provider=org.apache.hadoop.fs.auth.SimpleCredential
Provider \
-Dfs.cosn.userinfo.secretId=COS_SECRETID \
-Dfs.cosn.userinfo.secretKey=COS_SECRETKEY \
-Dfs.cosn.bucket.region=ap-guangzhou \
-Dfs.cosn.impl=org.apache.hadoop.fs.CosFileSystem \
-Dfs.AbstractFileSystem.cosn.impl=org.apache.hadoop.fs.CosN \
--src /data/warehouse \
--dest cosn://examplebucket-1250000000/warehouse
```

## What do I do if the result shows that some files failed to be copied?

COSDistCp retries the file copy process five times in case of IOException. If the copy still fails after five attempts, the information of the failed files will be written to the `/tmp/${randomUUID}/output/failed/` directory, where `${randomUUID}` is a random string. Common reasons for copy failure include:

1. The source file contained in the copy manifest is not found during the copy (recorded as SRC\_MISS).
2. The job initiator does not have permission to read the source file or write the destination file, or the copy failed due to other reasons (recorded as COPY\_FAILED).

If the log message indicates that the source file does not exist, and the source file is ignorable, you can run the following command to obtain the list of different files except for those recorded as SRC\_MISS:

```
hadoop fs -getmerge /tmp/${randomUUID}/output/failed/ failed-manifest
grep -v '"comment":"SRC_MISS"' failed-manifest |gzip > failed-
manifest.gz
```

Except for those recorded as SRC\_MISS, if there are other failed files, you can locate the failure reasons by referring to the error log messages in the `/tmp/${randomUUID}/output/logs/` directory and pulling the application logs. The following command example pulls the logs of the YARN application:

```
yarn logs -applicationId application_1610615435237_0021 >
application_1610615435237_0021.log
```

In the command above, `application_1610615435237_0021` is the application ID.

## Will COSDistCp generate incomplete files due to network or other exceptions?

If the network is abnormal, the source file is missing, or the permissions are insufficient, COSDistCp cannot generate a destination file with the same size as the source file.

- For versions earlier than COSDistCp 1.5, COSDistCp will attempt to delete the destination files generated. If the deletion fails, you need to re-execute the copy task to overwrite the incomplete files, or manually delete them.
- If your COSDistCp version is 1.5 or later and the version of the Hadoop-COS plugin in the running environment is 5.9.3 or later, when files fail to be copied to COS, COSDistCp will call the abort API to terminate the ongoing upload request. Therefore, no incomplete file will be generated even if an exception occurs.
- If your COSDistCp version is 1.5 or later but the version of Hadoop-COS in the running environment is earlier than 5.9.3, you are advised to upgrade it to 5.9.3 or later.
- If the destination location is not COS, COSDistCp will attempt to delete the destination files.

## There are some invisible incomplete multipart uploads in COS buckets, which occupy storage space. How do I deal with them?

COS buckets may have some incomplete multipart uploads occupying storage space due to incidents such as server exception and process kill. You can configure an incomplete multipart upload deletion rule as instructed in [Setting Lifecycle](#) to clear them.

## A memory overflow and task timeout occurred during migration. How do I adjust parameters?

During migration, both COSDistCp and the tools used to access COS and CHDFS, based on their own logic, occupy some memory. To avoid memory overflow and task timeout, you can adjust parameters of some MapReduce jobs, for example:

```
hadoop jar cos-distcp-${version}.jar -Dmapreduce.task.timeout=18000 -  
Dmapreduce.reduce.memory.mb=8192 --src /data/warehouse --dest  
cosn://examplebucket-1250000000/data/warehouse
```

As shown in the example above, the value of `mapreduce.task.timeout` is changed to 18,000 seconds to avoid job timeout when large files are copied, and the value of `mapreduce.reduce.memory.mb` (memory size of the Reduce process) is changed to 8 GB to avoid memory overflow.

## How do I control the migration bandwidth of the migration task through migration over Direct Connect?

The formula for calculating the total bandwidth limit of COSDistCp migration is:  $\text{taskNumber} \times \text{workerNumber} \times \text{bandWidth}$ . You can set `workerNumber` to 1, use the `taskNumber` parameter to control the number of concurrent migrations, and use the `bandWidth` parameter to control the bandwidth of a single concurrent migration.

# HDFS TO COS

Last updated: 2023-09-07 14:57:17

## Note

The HDFS to COS tool is designed to copy data from HDFS to Tencent Cloud Object Storage (COS).

## Operating Environment

### System environment

Linux or Windows.

### Software requirements

JDK v1.7 or v1.8

## Installation and configuration

For more information on environment installation and configuration, see [Java](#).

## Configuration Method

1. Install Hadoop v2.7.2 or higher. For detailed directions, see [Hadoop](#).
2. Download the HDFS TO COS tool from [GitHub](#) and decompress it.
3. Copy the `core-site.xml` file of the HDFS cluster to be synced to the `conf` folder. The `core-site.xml` file contains the configuration information of NameNode.
4. In the `cos_info.conf` configuration file, configure the bucket, region, and API key information. The bucket name is formed by connecting a user-defined string and the system-generated `APPID` with a hyphen, for example, `examplebucket-1250000000`.
5. Specify the configuration file location in the command line parameter. The default location is `conf/cos_info.conf`.

### Note

If the command line parameter conflicts with the configuration file, the command line parameter shall apply.

## How to Use

### Note

The following demonstrates how to use it with Linux as an example.

## Viewing help

```
./hdfs_to_cos_cmd -h
```

## Copying a file

- Copy from HDFS to COS. If a file with the same name as the file to be copied already exists in COS, the former will be overwritten.

```
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/
```

- Copy from HDFS to COS. If a file with the same name and length as the file to be copied already exists in COS, the latter will be skipped (this is suitable for repeated copy).

```
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/ -  
skip_if_len_match
```

Only the length is checked here, as the overheads would be very high if the digests of files in Hadoop are calculated.

- Copy from HDFS to COS. If the `Har` directory (Hadoop archive file) exists in HDFS, the `.har` files can be automatically decompressed by specifying the `--decompress_har` parameter:

```
./hdfs_to_cos_cmd --decompress_har --hdfs_path=/tmp/hive --  
cos_path=/hdfs/20170224/
```

If the `--decompress_har` parameter is not specified, the directory will be copied as an ordinary HDFS directory, that is, the files in the `Har` directory such as `index` and `masterindex` will be copied as-is.

## Directory information

```
conf: configuration file, which is used to store core-site.xml and cos_i  
nfo.conf  
log: log directory  
src: Java source program  
dep: compiled executable JAR package
```

## FAQs and Help

### Configuration information

Please make sure that the entered configuration information is correct, including bucket, region, and API key information. The bucket name is formed by connecting the user-defined string and system-generated `APPID` with a hyphen, such as `examplebucket-1250000000`. Please also make sure that the time on the server is in sync with Beijing time (if there is a difference of about 1 minute, it is okay, but if the difference is large, please set the server time correctly).

### DataNode

Please make sure that the server where the copy program is located can also access the DataNode. The NameNode uses a public IP address and can be accessed, but the DataNode where the obtained block is located uses a private IP address and cannot be accessed directly; therefore, it is recommended that the copy program be placed in a Hadoop node for execution, so that both the NameNode and DataNode can be accessed.

### Permissions

Please use the current account to download a file with the Hadoop command, check whether everything is correct, and then use the synchronization tool to sync the data in Hadoop.

### File overwriting

Files that already exist in COS will be overwritten by default in case of repeated upload, unless you explicitly specify the `-skip_if_len_match` parameter, which indicates to skip files during upload if they have the same length as existing files.

### cos path

`cos path` is considered as a directory by default, and files that are eventually copied from HDFS will be stored in this directory.

### Copying data from Tencent Cloud EMR HDFS

To copy data from Tencent Cloud EMR HDFS to COS, you are recommended to use the high-performance DistCp tool. For more information, please see [Data Migration Between Hadoop File System and COS](#).

# GooseFS-Lite

Last updated: 2025-08-06 11:29:54

## Feature Description

The GooseFS-Lite tool supports mounting [Cloud Object Storage \(COS\)](#) buckets locally, allowing you to directly operate objects in Tencent Cloud Object Storage as if using a local file system. Compared to the COSFS tool, GooseFS-Lite offers higher large file read and write speeds and is not limited by local disk performance. GooseFS-Lite supports major POSIX file system features, such as file order, random read, sequential write, and directory manipulation.

## Use Limits

**GooseFS-Lite is only suitable for simple management of files after mounting and does not support some features of local file systems.** Please note the following usage limits:

- Files cannot be written randomly and truncate operations are not supported.
- When multiple clients mount the same COS bucket, users need to self-coordinate the behavior of multiple clients, such as avoiding multiple clients writing to the same file.
- Rename operations for files/folders are non-atomic.
- Reading and renaming files being written in the current mount point are not supported.
- Metadata operations like listing directories have poor performance due to the need for remote access to the COS server.
- Soft/hard links are not supported.
- Append write performance is poor, involving server-side data copy and downloading the appended file.
- Not recommended for use in low memory scenarios, such as when container memory or CVM memory is less than 2G.
- Container environments are currently only supported within [Tencent Kubernetes Engine \(TKE\)](#). Non-TKE containers are not supported.
- Not recommended for use in scenarios with a large amount of random reads and high performance requirements.

### Note:

External network mount and append write operations on non-infrequent storage will incur download traffic fees.

## Usage Environment

- [KonaJDK 11](#)
- Linux X86\_64

## Usage

### Step 1. Install dependencies

#### CentOS/TencentOS Server

```
yum install -y fuse-devel
```

#### Ubuntu

```
apt install -y libfuse-dev
```

#### Other Linux distributions

#### Compile and install libfuse2.9.7

```
wget "https://github.com/libfuse/libfuse/releases/download/fuse-2.9.7/fuse-2.9.7.tar.gz"
tar xvf fuse-2.9.7.tar.gz
cd fuse-2.9.7
./configure
make -j8
make install
```

### Step 2: Installing GooseFS-Lite

Install GooseFS-Lite in the current directory and soft link `goosefs-lite` to `/usr/bin/goosefs-lite` for convenient use of the `goosefs-lite` command.

```
curl -fssL https://downloads.tencentgoosefs.cn/goosefs-lite/install.sh |
sh -x
cd goosefs-lite-*
sudo bash bin/install.sh
```

### Step 3: Installing KonaJDK11

In the `goosefs-lite-<specific version>` directory (for example, in the `goosefs-lite-1.0.6` directory), use the following command to install KonaJDK to `/usr/local/konajdk11`:

```
sudo bash bin/install-jdk.sh https://github.com/Tencent/TencentKona-11/releases/download/kona11.0.22/TencentKona-11.0.22.b1-jdk_linux-x86_64.tar.gz
```

As shown below, there are two options to choose from:

**First:** Use the download link for konajdk. **Second:** Download the konajdk installation package to the specified directory. Use the following command to install it, so that `goosefs-lite` can automatically use this Java runtime environment.

Usage:

Command: `install-jdk.sh http[s]://host/path`

Example: `install-jdk.sh https://github.com/Tencent/TencentKona-11/releases/download/kona11.0.22/TencentKona-11.0.22.b1-jdk_linux-x86_64.tar.gz`

or

Command: `install-jdk.sh /path/to/jdk.tar.gz`

Example: `install-jdk.sh /Downloads/TencentKona-11.0.22.b1-jdk_linux-x86_64.tar.gz`

If you want a more flexible way to install the Java environment, refer to [manual JDK installation](#) and modify environment variables in `conf/goosefs-env.sh` to make it effective.

## Step 4: Modifying the Configuration File

In the `goosefs-lite-<specific version>` directory (for example, in the `goosefs-lite-1.0.6` directory), there are two ways to modify the configuration file:

- Use `sed` to modify the following three parameters, filling in `SECRET_ID`, `SECRET_KEY`, and `REGION` as needed:
  - Set `fs.cosn.userinfo.secretKey` to the Tencent Cloud key.
  - Set `fs.cosn.userinfo.secretId` to the Tencent Cloud ID.
  - Set `fs.cosn.bucket.region` to the bucket region.

```
sed -i '/<name>fs.cosn.userinfo.secretId</name>/{N;s/<value>[<]*</value>/<value>$SECRET_ID</value>/' conf/core-site.xml
sed -i '/<name>fs.cosn.userinfo.secretKey</name>/{N;s/<value>[<]*</value>/<value>$SECRET_KEY</value>/' conf/core-site.xml
```

```
sed -i '/<name>fs.cosn.bucket.region</name>/{N;s/<value>[^<]*</value>/<value>$REGION</value>/' conf/core-site.xml
```

- You can also use vim to edit the conf/core-site.xml file to modify parameters.
  - Set fs.cosn.userinfo.secretKey to the Tencent Cloud key.
  - Set fs.cosn.userinfo.secretId to the Tencent Cloud ID.
  - Set fs.cosn.bucket.region to the bucket region.

## Configuration file description

In the `goosefs-lite-<specific version>/conf` directory (for example, in the `goosefs-lite-1.0.6/conf` directory), you can see the following files:

- **acl-site.properties:** specify directory permissions, username, group name, similar to Linux Posix semantics 0755, uid, gid, etc.
- **core-site.xml:** Hadoop-COS configuration file (goosefs-lite's data flow is based on Hadoop-COS, so parameters and configuration files are almost universal)
- **goosefs-env.sh:** various environment variables, such as JVM parameters, etc.
- **goosefs-lite.properties:** goosefs-lite configuration.
- **log4j.properties:** log configuration. To enable debug logging, uncomment the last line of this file and remount.
- **acl-site.properties:** specify directory permissions, username, group name, similar to Linux Posix semantics 0755, uid, gid, etc.

### Note:

- We recommend that users avoid using permanent keys in the configuration. Configuring sub-account keys or temporary keys can help improve business security. When authorizing a sub-account, grant only the permissions of the operations and resources that the sub-account needs, which helps avoid unexpected data leakage.
- If you must use a permanent key, we recommend that you limit ITS permission scope. This can improve the security by limiting its executable operations, resource scope, and conditions (access IP, etc.).

## Step 5: Mounting a Bucket to a Local Directory

Execute the following command in the `goosefs-lite-<specific version>` directory (for example, in the `goosefs-lite-1.0.6` directory) to mount the bucket configured in the key file to the specified directory:

```
./bin/goosefs-lite mount <MountPoint> cosn://<BucketName>/
```

Among them:

- <MountPoint> is the local mounting directory (for example, `/mnt/goosefs-lite-mnt-dir`), which must be empty; otherwise, it cannot be mounted.
- <BucketName> is the bucket name (e.g., `examplebucket-1250000000`).

**Example:**

```
mkdir -p /mnt/goosefs-lite-mnt
./bin/goosefs-lite mount /mnt/goosefs-lite-mnt/ cosn://examplebucket-
1250000000/
```

View the local mount point and the corresponding COS bucket. The output information includes the process ID, local mount point, and COS path in sequence:

```
$ ./bin/goosefs-lite stat
pid      mount_point          cos_path
13815    /mnt/goosefs-lite-mnt/ cosn://examplebucket-1250000000/
```

If you need to specify multiple mount parameters simultaneously in the command line, you can use commas to separate multiple parameters. The following command sets the mount point to read-only and allows other users to access the mount point:

```
./bin/goosefs-lite mount -o "ro,allow_other" mnt/ cosn://examplebucket-
1250000000/
```

Among them:

- `-o allow_other`: To allow other users to access the mount folder, you can specify this parameter when running GooseFS-Lite.
- `-o ro`: Set the mount point to read-only, disallowing write and delete operations.

**Note:**

A single parameter can be specified with `-o`, such as `-o ro`; multiple parameters can be separated by commas, such as `-o "ro,allow_other"`.

## Step 6: Uninstalling the Mount Point

In the `goosefs-lite-<specific version>` directory (for example, in the `goosefs-lite-1.0.6` directory), execute the following command to uninstall the mount point:

```
./bin/goosefs-lite umount /mnt/goosefs-lite-mnt
Unmount fuse at /mnt/goosefs-lite-mnt/ (PID: 17206).

# If uninstallation is abnormal, you can force uninstall using the
following command
$ sudo umount -l /mnt/goosefs-lite-mnt
```

## Step 7: Parameter Tuning

GooseFS-Lite includes two configuration files: `conf/core-site.xml` and `conf/goosefs-lite.properties`.

You can optimize upload and download bandwidth by modifying `conf/core-site.xml`. Common parameters are as follows, more parameters can be found in the [Hadoop-COS](#) documentation.

Attribute Key	Description	Default Value	Required
fs.cosn.useHttps	Configure whether to use HTTPS as the transfer protocol for the COS backend.	true	No
fs.cosn.upload.part.size	The size of each part in the chunked upload. Since COS can only support up to 10,000 parts in a chunked upload, it is necessary to estimate the maximum possible single file size. For example, when the part size is 8MB, a single file with a maximum size of 78GB can be uploaded. The maximum part size can be supported is 2GB, which means a single file can support up to 19TB.	8388608 (8MB)	No
fs.cosn.upload_thread_pool	The number of concurrent threads when files are uploaded to COS through streams.	32	No
fs.cosn.read.ahead.block.size	The size of the pre-read block.	1048576 (1MB)	No
fs.cosn.read.ahead.queue.size	The length of the read-ahead queue.	6	No

You can adjust the behavior of GooseFS-Lite by modifying `conf/goosefs-lite.properties`. Common parameters are as follows:

Attribute	Description	Default Value	Required
goosefs.fuse.list.entries.cache.enabled	Enable client List cache	true	No
goosefs.fuse.list.entries.cache.max.size	Maximum number of entries in client List cache, unit: entries	100000	No
goosefs.fuse.list.entries.cache.max.expiration.time	Valid time of client List cache, unit: ms	15000	No
goosefs.fuse.async.release.max.wait.time	Time to wait for write operation to complete when files in open and rename operations are being written, unit: ms	5000	No
goosefs.fuse.umount.timeout	Time to wait for incomplete operations when uninstalling the file system, unit: ms	120000	No

When your read and write concurrency is high, you can adjust the maximum JVM runtime memory of GooseFS-Lite as follows to avoid FullGC and OutOfMemoryError. The default JVM values are `-Xmx512m -XX:MaxDirectMemorySize=512m -XX:+UseG1GC -XX:G1HeapRegionSize=32m`. The adjustment method is as follows:

```
export JAVA_OPTS="-Xms2G -Xmx2G"
./bin/goosefs-lite mount /mnt/goosefs-lite-mnt/ cosn://examplebucket-1250000000/
ps -ef|grep goosefs-lite|grep -v grep
```

## FAQs

### Missing libfuse library file, how to handle it?

```
java.lang.UnsatisfiedLinkError: /tmp/libjnihfuse2359305659114946750.so: libfuse.so.2: cannot open shared object file: No such file or directory
    at java.base/java.lang.ClassLoader$NativeLibrary.load0(Native Method)
    at java.base/java.lang.ClassLoader$NativeLibrary.load(ClassLoader.java:2445)
    at java.base/java.lang.ClassLoader$NativeLibrary.loadLibrary(ClassLoader.java:2501)
    at java.base/java.lang.ClassLoader.loadLibrary0(ClassLoader.java:2700)
    at java.base/java.lang.ClassLoader.loadLibrary(ClassLoader.java:2630)
    at java.base/java.lang.Runtime.load0(Runtime.java:768)
    at java.base/java.lang.System.load(System.java:1837)
    at com.qcloud.cos.goosefs.jnihfuse.utils.NativeLibraryLoader.loadLibraryFromJar(NativeLibraryLoader.java:105)
    at com.qcloud.cos.goosefs.jnihfuse.utils.NativeLibraryLoader.loadLibrary(NativeLibraryLoader.java:83)
    at com.qcloud.cos.goosefs.jnihfuse.LibFuse.loadLibrary(LibFuse.java:48)
    at com.qcloud.cos.goosefs.jnihfuse.LibFuse.<clinit>(LibFuse.java:32)
    at com.qcloud.cos.goosefs.jnihfuse.AbstractFuseFileSystem.<clinit>(AbstractFuseFileSystem.java:41)
    at com.qcloud.cos.goosefs.fuse.GooseFSLiteFuse.main(GooseFSLiteFuse.java:64)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.base/java.lang.reflect.Method.invoke(Method.java:566)
    at org.springframework.boot.loader.MainMethodRunner.run(MainMethodRunner.java:53)
    at java.base/java.lang.Thread.run(Thread.java:829)
```

To install libfuse, follow these steps:

### Method 1

1. Run the following command to install fuse-devel.

- If you are using CentOS or TencentOS systems, run the following command:

```
yum install fuse-devel
```

- If you are using Ubuntu systems, run the following command:

```
apt install libfuse-dev
```

2. After installation, run the following command to check if the installation was successful.

```
find / -name libfuse.so*
```

### Method 2

Update the old version libfuse.so.2.9.2. The installation steps are as follows:

**Note:**  
CentOS 7 installs libfuse.so.2.9.2 by default.

1. Download the [libfuse source code](#) and compile and generate libfuse.so.2.9.7.

```
tar -zxvf fuse-2.9.7.tar.gz
cd fuse-2.9.7/ && ./configure && make && make install
echo -e '\n/usr/local/lib' >> /etc/ld.so.conf
ldconfig
```

2. After compiling and generating `libfuse.so.2.9.7`, follow these steps to replace it:

2.1 Run the following command to find the old version `libfuse.so.2.9.2` library links.

```
find / -name libfuse.so*
```

2.2 Run the following commands to copy `libfuse.so.2.9.7` to the location of the old version library `libfuse.so.2.9.2`.

```
cp /usr/local/lib/libfuse.so.2.9.7 /usr/lib64/
```

2.3 Run the following commands to delete all links of the old version `libfuse.so` library.

```
rm -f /usr/lib64/libfuse.so
rm -f /usr/lib64/libfuse.so.2
```

2.4 Run the following commands to build `libfuse.so.2.9.7` library links similar to those of the old version library.

```
ln -s /usr/lib64/libfuse.so.2.9.7 /usr/lib64/libfuse.so
ln -s /usr/lib64/libfuse.so.2.9.7 /usr/lib64/libfuse.so.2
```

## Configuring Boot Mount

1. Edit the file `/usr/lib/systemd/system/goosefs-lite.service` and add the following content. You can replace `examplebucket-1250000000` with your bucket: Note that the memory values configured in `JAVA_OPTS` (`-Xms` and `-Xmx`) should not exceed 50% of the physical memory limit of the node. For example, if the node has 16GB of physical memory, it is recommended to configure up to `-Xms8G -Xmx8G`.

The following uses version `goosefs-lite-1.0.6` as an example:

```
[Unit]
Description=The Tencent Cloud GooseFS Lite for COS
Requires=network-online.target
After=network-online.target

[Service]
Type=forking
User=root
```

```
Environment="JAVA_OPTS=-Xms2G -Xmx4G -XX:MaxDirectMemorySize=1G -
XX:+UseG1GC -XX:G1HeapRegionSize=32m"
ExecStart=/usr/local/goosefs-lite-1.0.6/bin/goosefs-lite mount
/mnt/goosefs-mnt cosn://examplebucket-1250000000/
ExecStop=/usr/local/goosefs-lite-1.0.6/bin/goosefs-lite umount
/mnt/goosefs-mnt
Restart=always
RestartSec=5

[Install]
WantedBy=multi-user.target
```

2. Run the following command to execute the mount command and view the status of the background Daemon process.

```
# Apply the systemd configuration of goosefs-lite
systemctl daemon-reload
# Start the background Fuse process
systemctl start goosefs-lite
# View the status of the background Daemon process
systemctl status goosefs-lite
# View the mount point list
/usr/local/goosefs-lite-1.0.6/bin/goosefs-lite stat
# If modifying the systemd configuration, reload and restart after
modification.
```

### Set to attempt mounting at boot startup:

```
systemctl enable goosefs-lite
```

3. Uninstall the mount point, reboot the machine, and check the status of the Fuse process.

```
# Execute Uninstall. Note: Do not Uninstall during data writing, as it
may cause data incomplete.
systemctl stop goosefs-lite
# Restart operating system. Proceed with caution to avoid affecting
business.
reboot -h now
# View the status of the background Daemon process
```

```
systemctl status gosefs-lite
# View the mount point list
/usr/local/gosefs-lite-1.0.6/bin/gosefs-lite stat
```

## GooseFS-Lite Having High CPU Utilization and Sending a Large Number of HEAD and LIST Requests to COS, Incurring a Large Amount of Request Fees During a Certain Period Every Day, What Should I Do

This is usually caused by a scheduled disk scan task on your machine. The common disk scan program on Linux systems is updatedb. You can add the GooseFS-Lite mount point directory to the PRUNEPATHS configuration item in the updatedb configuration file `/etc/updatedb.conf` to avoid the disk scan behavior of this program. Additionally, you can use the Linux tool `auditd` to find the programs accessing the GooseFS-Lite mount point.

Directions are as follows:

### 1. Install `auditd`.

- If you are using Ubuntu systems, run the following command:

```
apt-get install auditd -y
```

- If it is a CentOS system, execute the following command:

```
yum install audit audit-libs
```

### 2. Start the `auditd` service.

```
systemctl start auditd
systemctl enable auditd
```

### 3. Monitor the mounted directory.

#### Note:

`-w` specifies the GooseFS-Lite mounted directory, `-k` indicates the key to be outputted to the audit logs.

```
auditctl -w /usr/local/service/mnt/ -k gosefs_lite_mnt
```

### 4. Determine the access program based on the logs.

The audit log directory: `/var/log/audit`, the query command is as follows:

```
ausearch -i|grep 'goosefs_lite_mnt'
```

## 5. Stop auditd service.

If you need to stop the auditd service, you can use the following command:

```
/sbin/service auditd stop
```

### Note:

If the program accessing the mount point is always running, the newly started auditd will not monitor the access behavior of the program. This is because only the first call in multiple calls from the program to the mounted directory will be recorded.

## Handling "Cannot Allocate Memory" Error During GooseFS-Lite Installation

```
OpenJDK 64-Bit Server VM warning: INFO: os::commit_memory(0x0000000080000000, 2147483648, 0) failed; error='Cannot allocate memory' (errno=12)
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map 2147483648 bytes for committing reserved memory.
# An error report file with more information is saved as:
# /usr/goosefs-lite/goosefs-lite-1.0.2/hs_err_pid2860293.log
~
```

This error mainly occurs because of a memory allocation error during the GooseFS-Lite operation, usually when the requested memory exceeds the actual available memory. You can modify the `JAVA_OPT` parameter in the `./bin/goosefs-lite` file to a reasonable memory value, ensuring that the requested memory amount is less than the instance available memory amount.

## Viewing the Latest Package Version

Run the following command, the return value is the latest version number.

```
curl -fsSL https://downloads.tencentgoosefs.cn/goosefs-lite/LATEST_VERSION
```

## Viewing Logs

Problem troubleshooting relies on logs. Below are the locations of the relevant logs.

- For **goosefs-lite version 1.0.3 and later**, the default log directory is under `/data/goosefs/logs/fuse`.

For example: If the user mounting **goosefs-lite** is **root** and the mount point path is `/data1/data2`, then the log path is: `/data/goosefs/logs/fuse/root/data1/data2`.

- For **goosefs-lite versions 1.0.0–1.0.2**, the default log directory is under `/data/goosefs/logs`.

Another way to check the mount point log path is to first remount, then `ps aux | grep ${MOUNT_POINT}`. From the output, you can see `Error_File` or `goosefs.logs.dir`, and the parent directory of this path is where all logs for the mount point are located.

System logs: If it is a centos or tlinux system, it is `/var/log/message*`. If it is ubuntu, it is `/var/log/syslog`.

If you need to enable debug logs, go to `conf/log4j.properties`, comment out the last line, and remount to take effect.

#### Note:

Enabling debug logs will impact performance. Normally, there is no need to enable them.

## Mount failed with error "Name or service not known", how to handle it?

```
2023-05-16 19:23:20,668 [Thread-8] ERROR jnifuse.AbstractFuseFileSystem (AbstractFuseFileSystem.java:statfsCallback) -
Failed to statfs /:
java.lang.RuntimeException: java.net.UnknownHostException: VM-36-104-centos: VM-36-104-centos: Name or service not known
    at com.qcloud.cos.goosefs.util.network.NetworkAddressUtils.getLocalIpAddress(NetworkAddressUtils.java:522)
    at com.qcloud.cos.goosefs.util.network.NetworkAddressUtils.getLocalHostName(NetworkAddressUtils.java:444)
    at com.qcloud.cos.goosefs.util.network.NetworkAddressUtils.getLocalHostMetricName(NetworkAddressUtils.java:463)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.constructSourceName(MetricsSystem.java:202)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.lambda$static$0(MetricsSystem.java:89)
    at com.qcloud.cos.goosefs.util.CommonUtils$.firstTime(CommonUtils.java:799)
    at com.qcloud.cos.goosefs.util.CommonUtils$.get(CommonUtils.java:794)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.getMetricNameWithUniqueId(MetricsSystem.java:390)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.lambda$getClientMetricName$5(MetricsSystem.java:341)
    at java.util.concurrent.ConcurrentHashMap.computeIfAbsent(ConcurrentHashMap.java:1660)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.getClientMetricName(MetricsSystem.java:340)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.getMetricName(MetricsSystem.java:271)
    at com.qcloud.cos.goosefs.metrics.MetricsSystem.timer(MetricsSystem.java:529)
    at com.qcloud.cos.goosefs.fuse.GooseFSFuseUtils.call(GooseFSFuseUtils.java:287)
    at com.qcloud.cos.goosefs.fuse.GooseFSLiteJniFuseFileSystem.statfs(GooseFSLiteJniFuseFileSystem.java:1091)
    at com.qcloud.cos.goosefs.jnifuse.AbstractFuseFileSystem.statfsCallback(AbstractFuseFileSystem.java:289)
```

This is usually because the domain name cannot be resolved. You can try pinging the domain name. If the error shown above occurs, you can execute the following command:

```
ping VM-36-104-centos
```

If it also returns a failure, you can configure the corresponding IP by modifying `/etc/hosts`. Generally, you can set it to `127.0.0.1`.

**Follow the steps below:**

1. Add a line in the `/etc/hosts` file. Replace `VM-36-104-centos` with your hostname.

```
127.0.0.1 VM-36-104-centos
```

2. Then retest with ping. After confirming normal resolution, remount to take effect.

## Default Environment JDK Is Not KonaJDK11, How to Use GooseFS-lite

1. Download the [KonaJDK11](#) package and extract it.
2. Copy the absolute path of the java binary program in konajdk. For example, the extracted jdk is under `/root/konajdk11`.

```
# Determine Java Version
/root/konajdk11/bin/java -v
# Modify the Current Shell's Java Environment Variable Without
Affecting Other Shells and Processes
export JAVA=/root/konajdk11/bin/java
# Mounting
goosefs-lite mount /mnt cosn://bucket-appid
```

3. Using `ps aux | grep goosefs-lite`, you can see the process starts with `/root/konajdk11/bin/java`, indicating that the specified java version is being used, operation completed.

## The Mount Point Was Originally Normal, Suddenly Unable to Use During Operation, How to Handle

Assume the current problematic mount point is `/tmp/mount_point`.

1. First, use `ps aux | grep /tmp/mount_point` to check if any process is using this mount point, including `goosefs-lite`. If there is, use the kill command to terminate the corresponding process.
2. Use `ls` to attempt to access the mount point. If the return is empty, it means the mount point has been successfully unmounted. Then remount to take effect.

```
ls /mount_point
```

3. If an error like **transport is not connected** is thrown, execute `umount -l /mount_point` to force unmount. (This command requires root privileges.)

Usually, such situations are caused by processes being killed by `kill -9` or the system oom-killer. You can find it in the system logs or `goosefs-lite` logs.

4. Check the mount point log directory (`/data/goosefs/logs/fuse/$USER/$MOUNT_POINT`) for logs starting with `hs_error` (file describes the stack and reason before program exit).

## Throwing Exception: Unsupported or Unrecognized SSL Message, how to handle it?

The current environment does not support disabling https mode, configuration needs to be modified. The solution is to add the following content to the `core-site.xml` configuration file:

```
<property>
  <name> fs.con.useHttps </name>
  <value> false </value>
</property>
```

## How to Access COS Using Intranet Domain Name in GooseFS-Lite

In the `core-site.xml` configuration file, delete the `fs.cosn.bucket.region` property and add `fs.cosn.bucket.endpoint_suffix`.

To learn more about parameters, read the following documents:

- [GitHub – tencentyun/hadoop-cos](#)
- [Object Storage Hadoop Tools](#)

## Handling 403 Forbidden During Mounting or Usage

Typically, ERROR logs in the log will describe what permissions are missing, supplement as needed.

Note: For head bucket permissions, it needs to be set at the bucket level rather than just the path level. For example, `cosn://bucket-appid/path` will not work, it must be `cosn://bucket-appid`. This permission will not expose the objects in the bucket.

## Handling Error "fuse: failed to open /dev/fuse Operation not permitted"

1. Check if you have root privileges.
2. If in a container, check if the container is started with `--privileged`. If not, add it.
3. Check if the fuse kernel module is installed: execute the command `lsmod | grep fuse` and see if there is any output. If not, it indicates that the fuse kernel module is missing.

## Handling File Write Failure with Error "part num: 10001, the parameter partNumber is not valid"

This indicates that the number of parts for multipart upload exceeds the limit. COS supports a maximum of 10,000 parts. Therefore, `goosefs-lite` by default supports files up to  $8\text{MB} \times 10000$  (approximately 78GB). If you need to support larger files, you need to adjust the parameter `fs.cosn.upload.part.size`. For example, if `fs.cosn.upload.part.size` is changed to 16777216 (i.e., 16MB), it can support large files up to  $16\text{MB} \times 10000$ .

# Online Tools

## COS Request Tool

Last updated: 2023-09-07 11:06:11

### Feature Overview

The COS request tool is a web-based debugging tool provided by COS. It is integrated on the TencentCloud API 3.0 Explorer platform for API debugging.

#### Note

The requests sent by the COS request tool will be genuinely sent to the COS business server. **All actions are equivalent to real operations, so please exercise caution when choosing DELETE-type operations.**

The COS request tool supports XML APIs but not JSON APIs.

- JSON APIs were provided by COS for you to access COS before XML APIs were released. Both types of APIs have the same underlying architecture where data is interconnected; however, they are incompatible with each other.
- XML APIs have a richer set of features and strengths over JSON APIs. We strongly recommend you upgrade to XML APIs for COS.

### Tool URL

Click [here](#) to enter the COS request tool.

### How to Use

Search for the **Object Storage** product at the top of the interface, select the desired API, fill in the corresponding parameters for the selected API, and click **Send Request** to obtain the response results.

The COS request tool's interface consists of three sections from left to right: the API section, the parameter section, and the result section. You can perform corresponding actions in each section, and ultimately send requests and obtain response results and related process parameter information in the result section.

The detailed steps to use the COS request tool are as shown below:

#### 1. Select the COS product.

Search for **Object Storage** at the top of the interface to view the related COS API endpoints.

#### Note

The COS request tool is integrated on the TencentCloud API 3.0 platform that provides API debugging tools for many Tencent Cloud products. You can also select other products to debug their APIs as needed.

### Select the API interface you want to debug

You can choose the appropriate API for debugging based on your needs. The API interface section displays various COS-related APIs, such as Service interfaces, Bucket class interfaces, and Object class interfaces.

- In Service class interfaces, such as GET Service, this API lists all bucket information under your account. You need to enter your API key information. To retrieve bucket information for your account in a specific region, select the corresponding region in the parameter column. For more details about this API, please refer to the [GET Service](#) documentation.
- In the Bucket category, there are API calls for operating on buckets, such as PUT Bucket lifecycle. For more information on Bucket APIs, please refer to [Bucket API calls](#).
- In the Object category, there are various APIs for object operations, such as PUT Object. For more information on Object APIs, please see [Object Interface](#).

For more information on the available APIs, see [Operation List](#).

### 3. Enter parameters for the API.

The parameter section lists the corresponding parameters for the selected API. For more information on the parameters of COS APIs, see [API Documentation](#).

API key information is a required parameter when calling API interfaces. When using API interfaces to operate on resources such as buckets or objects, you need to enter API key information to authorize the API request. It is recommended to use sub-account keys, following the principle of least privilege to reduce risks. For obtaining sub-account keys, refer to [Sub-account Access Key Management](#).

#### Note

For each API, the COS request tool displays a red asterisk behind each required parameter. You can also select **Only Required Parameters** to view required parameters only in the parameter section.

### 4. Send a request and view the response.

After selecting an API and entering parameters, click **Send Request** on the **Online Call** tab. Your request will be sent to the server, and the server will manipulate your buckets or objects accordingly.

#### Note

Requests sent by the COS request tool will be sent to the real COS server. **As all operations are real, be careful when performing operations such as `DELETE`.**

After the request is sent, the returned result and the request parameters will be displayed in the result section. The **Request Parameters** part lists your HTTP request body; the **Response Result** part lists the response body of the request; the **Signature Process** part lists the signature involved in the request and its generation process; and the **Curl** part lists the statement called by Curl.

#### Example:

For example, a `GET Object` request is sent to get a file named `0001.txt` as shown below. The **Request Parameters** part lists the corresponding parameters of the request:

```
GET https://bucketname-appid.cos.ap-region.myqcloud.com/0001.txt
Host: bucketname-appid.cos.ap-region.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-
ak=AKIDwqaGoCIWIG4hDwdJUTL5e3hn04xi**&q-sign-
time=1543398166;1543405366&q-key-time=1543398166;1543405366&q-header-
list=host&q-url-param-list=&q-
signature=f50ddd3e0b54a92df9d4efe2d0c3734a8c90**
```

The first line displays the HTTP Verb and the accessed link, the second line shows the accessed domain, and the last line displays the signature information for this request. For PUT-type requests, the request header information is more complex, but there are still some common request headers. For information about common request headers, please see [Common Request Headers](#).

In the **Signature Process**, you can view the signature involved in this request and its generation process. For a detailed introduction to the signature algorithm, please refer to [Request Signature](#). If you need to generate and debug request signatures, we recommend using the [COS Signature Tool](#).

The response result returned by COS is as follows:

```
200 OK
content-type: text/plain
content-length: 6
connection: close
accept-ranges: bytes
date: Wed, 28 Nov 2018 09:42:49 GMT
etag: "5a8dd3ad0756a93ded72b823b19dd877"
last-modified: Tue, 27 Nov 2018 20:05:26 GMT
```

```
server: tencent-cos
x-cos-request-id: NWJmZTYzMTlfOWUxYzBiMDlfOTA4NF8yMWI2****
x-cos-version-id: MTg0NDY3NDI1MzAzODkyMjU****
hello!
```

The first line, "200 OK," represents the status code information for the request. If the request fails, an appropriate error code will be returned. For more information on error codes, please see the [Error Codes](#) documentation. The following lines contain response header information. Different APIs have varying response bodies, but some common response header information exists. For more details on common response headers, please refer to [Common Response Headers](#).

## Supports and Limits

When you click **Send Request** and confirm to send the request with the required parameters to the COS server, COS will perform the corresponding actions on your bucket and objects. These actions cannot be undone or rolled back, so please exercise caution.

# Policy Generator

Last updated: 2025-08-29 10:38:56

## Feature Overview

The COS Policy Generator Tool is a web-based utility provided by Tencent Cloud Object Storage (COS), designed to offer users a graphical interface for effortless configuration and one-click generation of JSON-formatted policies.

After configuring the relevant parameters for a given policy type, the system generates ready-to-use JSON policy syntax for seamless backend processing.

Currently, the tool supports three types of JSON policies: User Policies, Bucket Policies, and STS Policies.

For an overview and detailed information regarding each policy type, please refer to [User Policies \(CAM Policies\)](#), [Bucket Policies](#), and [Accessing COS with Temporary Credentials](#).

## Supports and Limits

- Policy generator tool does not validate or notify users of incorrect parameters entered. To ensure the generated JSON policy takes effect, users must manually verify all field entries align with their specific business requirements.
- Policy Generator validates your inputs and flags invalid parameters, but cannot guarantee the generated policy will be operationally effective.
- Policy Generator validates mandatory fields – it detects, alerts, and prevents generation until all required parameters are complete.

## Tool URL

1. You can directly access the page by clicking on the COS Policy Generator Tool.
2. Alternatively, visit the page through the [COS Console](#). For detailed steps, refer to the Operation Steps.

## Applicable Scenarios

The COS Policy Generator Tool is suitable for the following scenarios:

- Users who are familiar with user policies, bucket policies, and STS policies but lack proficiency in JSON syntax.
- Users who need to add user policies, bucket policies, or STS policies via [API](#) or [SDK](#).
- Users who require batch additions of user policies, bucket policies, or STS policies.

## User Policy (CAM Policy)

**CAM (Cloud Access Management)** allows you to define customized access permissions for different users under your primary account. These access controls are implemented through policy syntax and apply permissions per user, defining these as User Policies. For more information, please see [User Policy](#) .

### Note:

- User policy must be written and completed before associating them with sub-users, user groups, or roles.
- User policy do not support granting operation and resource permissions to anonymous users.

When concerned about user actions, use user policies. Identify CAM users and check permissions of their user groups to understand capabilities. Recommended scenarios include:

- Configuring service-level permissions for Cloud Object Storage (COS), such as creating buckets (PutBucket) or listing buckets (GetService).
- Utilizing all COS buckets and objects under the primary account.
- Granting identical permissions to a large number of CAM users under the primary account.

## Bucket Policy

Bucket policy applies to configured buckets and the objects within them. Through bucket policies, you can authorize CAM sub-accounts, other primary accounts, or even anonymous users to perform operations on buckets and their objects. For more information, please see [Bucket Policy](#) .

When you are concerned about who can access a specific COS bucket, it is recommended to use bucket policies. Locate the bucket and review its policy to understand access permissions. Recommended scenarios include:

- Granting permissions for a specific bucket.
- More customizable than ACL
- Supporting cross-account and anonymous user authorization, (user policies cannot provide this).

## STS Policy

STS policy is associated with temporary credentials. Temporary credentials are temporary access tokens provided by the [Security Token Service \(STS\)](#). During the application process for temporary credentials, you can use the Policy parameter (i.e., STS policy) to restrict operations and resources, thereby adding temporary policy constraints to limit the permissions of the credential user. For more details, please refer to the [Using Temporary Credentials to Access COS](#), [Temporary Credential Generation and Usage Guide](#), and [Temporary Credential Security Guide for Front-End Direct Upload to COS](#).

**Note:**

Before applying for temporary credentials, you must have a Cloud Access Management (CAM) user (Tencent Cloud primary account or sub-account).

Recommended scenarios for temporary credentials include:

- Granting third-party temporary access to COS. For instance, if a user develops a client app that stores data in a COS bucket, storing permanent credentials directly in the app client is insecure. However, temporary credentials can be used to grant upload and download permissions.
- Temporary authorization for front-end data direct uploads. Refer to practices such as [Web Direct Upload](#), [Mini Program Direct Upload](#), [Mobile Application Direct Upload](#), [Flutter Direct Upload](#), and [uni-app Direct Upload](#) for using temporary credentials.

## Operational Steps

1. Log in to the [COS Console](#).
2. In the left navigation bar, click **Common Tools > Policy Generator Tool**.
3. On the Policy Generator Tool page, select the policy type and fill in the parameters according to the instructions provided.

**Note:**

- When granting permissions, ensure that you adhere to the principle of [least privilege](#), restricting users to perform only necessary operations (e.g., granting read-only permissions) and accessing resources with specific prefixes.
- If you directly grant other users permissions for all resources (``resource:*``) or all operations (``action:*``), the overly broad scope of permissions may lead to unintended privilege escalation, posing data security risks.

CAM Policy

[Signature Tool](#)   [Get Signature URL](#)   **[Policy Generator Tool](#)**   [Request Tool](#)   [Diagnostic Tool](#)

When authorizing, it is recommended to strictly adhere to the [Principle of Least Privilege](#) . Limit user access to specific operations (such as read-only permissions) and define resource prefixes to prevent granting unnecessary privileges, which could result in unintended privilege escalation and pose security risks to data.

Policy type \*    CAM Policy    Bucket Policy    STS Policy

Effect \*    Allow    Deny

Resource Path \*

Region	Bucket Name	Specific Path	Ope...
Empty List			
<a href="#">Add Operation</a>			

Operation \*

Operation Name	Operation
Empty List	
<a href="#">Add Operation</a>	

Condition

Condition Key	Condition Oper...	Condition Value	Operation
Empty List			
<a href="#">Add Condition</a>			

[Generate Policy](#)

**Result Feedback** [Clear All](#)

JSON Policy

```
{}
```

1. On the current page, select CAM Policy as the policy type.
2. Parameter field descriptions are provided below. For more details about configuration items, please refer to the [CAM Authorization Guide](#) , [CAM Policy Syntax](#) , and [Access Policy Language Overview](#) .

Parameter	Required(Yes or No)	Attribute	Description
Effect	Yes	effect	Support select "Allow" or "Deny", corresponding to "Allow" and "Deny" in the policy syntax.
Resource Path	Yes	Resource	<p>The resource path consists of region, bucket name, and specific path, and supports adding either an entire bucket or a specified resource.</p> <ul style="list-style-type: none"> <li>● Entire bucket: If you want to define the resource scope as an entire bucket, only the region and bucket name need to be filled in.</li> <li>● Specified resource: If you want to limit the resource scope to a specific folder, the region, bucket name, and specific path must be provided.</li> </ul> <div style="border: 1px solid #00aaff; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> For details such as Region, Bucket Name, and Specific Path, please refer to the descriptions of the corresponding parameters in the current table for guidance.</p> </div>
Region	Yes	/	The region where the bucket is located, for example, ap-beijing. For more information about regions, please refer to <a href="#">Regions and Endpoints</a> .
Bucket Name	Yes	/	The bucket name consists of a custom name and an APPID, for example: examplebucket-1250000000. For more information about bucket naming and APPID, please refer to <a href="#">Bucket Naming Conventions</a> , <a href="#">APPID</a> , and <a href="#">Glossary of Object Storage API Terms</a> .
			<p>Specific Resource Path If no specific path is specified, the default value is *.</p>

Specific Path	Yes	/	<ul style="list-style-type: none"> <li>● If you need to specify a particular path, please fill in the field. Examples: To target all objects in the /folder/ directory: enter folder/*</li> <li>● To target the specific object /folder/exampleobject: enter folder/exampleobject</li> <li>● To target all objects with the .txt extension in the bucket: enter *.txt</li> </ul>
Operation	Yes	Action	Add or remove operations you wish to authorize. For more information about operations, please refer to the <a href="#">Operation List</a> .
Condition	No	Condition	Specify conditions when granting permissions, for example, restricting the user's access IP. For more information about conditions, please refer to <a href="#">Condition Keys Description and Usage Examples</a> .

## Bucket Policy

Signature Tool   Get Signature URL   **Policy Generator Tool**   Request Tool   Diagnostic Tool

When authorizing, it is recommended to strictly adhere to the [Principle of Least Privilege](#) , Limit user access to specific operations (such as read-only permissions) and define resource prefixes to prevent granting unnecessary privileges, which could result in unintended privilege escalation and pose security risks to data.

Policy type \*    CAM Policy    Bucket Policy    STS Policy

Effect \*    Allow    Deny

User \*

User Type	Account ID	Operation
All Users	*	Delete
<a href="#">Add User</a>		

Resource \*    Entire Bucket    Specified Resource

Policy ID   Only English characters are supported. if left blank, a random string will be generated.

Region \*   Enter the bucket region, e.g., ap-guangzhou.

Bucket Name \*   Enter the bucket name, e.g., examplebucket-1250000000.

Resource Path \*   Enter the resource path.

Operation \*

Operation Name	Operation
All Operations	Delete
<a href="#">Add Operation</a>	

Condition

Condition Key	Condition Oper...	Condition Value	Operation
Empty List			

[Generate Policy](#)

**Result Feedback**   [Clear All](#)

JSON Policy

1. On the current page, select Bucket Policy as the policy type.
2. Parameter descriptions are provided below. For further details about configuration items, please refer to [Adding Bucket Policy](#) and [Access Policy Language Overview](#).

Parameter	Required( Yes or No)	Attribute	Description
Effect	Yes	Effect	Support select "Allow" or "Deny", corresponding to "Allow" and "Deny" in the policy syntax.
User	Yes	Principal	<p>Supports adding or removing authorized users, including: All Users (*), Root Account, Sub-account, and Cloud Service.</p> <p>Configuration details are as follows:</p> <ul style="list-style-type: none"> <li>● All Users: Select this option if you wish to grant operational permissions to anonymous users. Once configured, requests for objects will not require signatures. All users will be able to access objects directly via links, which poses a data leakage risk. Please configure with caution.</li> <li>● Root Account: Select this option if you wish to grant operational permissions to a specific root account. After selecting, you must specify the unique UIN of the root account.</li> <li>● Sub-account: Select this option if you wish to grant operational permissions to a specific sub-account. After selecting, you must specify the unique UIN of the sub-account.</li> <li>● Cloud Service: Select this option if you wish to grant operational permissions to a specific cloud service. After selecting, you must specify the exact cloud service.</li> </ul>
Resource	..	Resource	<p>Resources are described in a six-segment format and support specifying either an entire bucket or a specific resource.</p> <p>Configuration details are as follows:</p> <ul style="list-style-type: none"> <li>● Entire Bucket: Select this option if you need to configure bucket-level permissions or define the resource scope as the entire bucket.</li> <li>● Specific Resource: Select this option to limit the resource scope to a specific folder. After selecting this, you must further specify the exact directory in the "Resource Path"</li> </ul>

e	Yes	e	field.
			<div style="border: 1px solid #00aaff; padding: 10px;"> <p><b>Note:</b> After selection, information such as Region, Root Account UIN, Bucket Name, and Resource Path must be filled in. You may refer to the descriptions of the corresponding parameters in the current table for guidance.</p> </div>
Policy ID	No	Sid	The ID of the current policy rule, which supports only English characters. If this field is not explicitly specified, a random string will be automatically generated.
Region	Yes	/	The region where the bucket is located, for example, ap-beijing. For more information about regions, please refer to <a href="#">Regions and Access Endpoints</a> .
Root Account UIN	Yes	/	If the selected user type is Sub-account or Cloud Service, the Root Account UIN must be specified. For details about the UIN concept, please refer to <a href="#">UIN</a> .
Bucket Name	Yes	/	The bucket name consists of a custom name and an APPID, for example: examplebucket-1250000000. For more information about bucket naming and APPID, please refer to <a href="#">Bucket Naming Conventions</a> , <a href="#">APPID</a> , and <a href="#">Glossary of Object Storage API Terms</a> .
Resource Path	Yes	/	<p><b>Specific Resource Path</b></p> <p>If you select "Entire Bucket" as the resource type, the resource path will default to the entire bucket.</p> <p>If you select "Specific Resource" as the resource type, you must provide a detailed resource path. Examples:</p> <ul style="list-style-type: none"> <li>● To target all objects within the /folder/ directory: enter folder/*</li> <li>● To target the specific object /folder/exampleobject: enter folder/exampleobject</li> </ul>

			<p>to: <code>exampleobject</code></p> <ul style="list-style-type: none"> <li>● To target all objects with the <code>.txt</code> extension in the bucket: enter <code>*.txt</code></li> </ul>
Operation	Yes	Action	<p>Add or remove operations you wish to authorize. For more information about supported actions, please refer to the <a href="#">Operations List</a>.</p> <div style="border: 1px solid #00aaff; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> CDN service accounts only support the following operations such as <code>GetObject</code>, <code>HeadObject</code>, <code>OptionsObject</code></p> </div>
Condition	No	Condition	<p>When granting permissions, you may specify conditions such as restricting a user's access IP. For more details about conditions, please refer to .</p>

**STS Policy**

Signature Tool   Get Signature URL   **Policy Generator Tool**   Request Tool   Diagnostic Tool

When authorizing, it is recommended to strictly adhere to the [Principle of Least Privilege](#), Limit user access to specific operations (such as read-only permissions) and define resource prefixes to prevent granting unnecessary privileges, which could result in unintended privilege escalation and pose security risks to data.

Policy type \*    CAM Policy    Bucket Policy    STS Policy

Effect \*    Allow    Deny

Resource Path \*

Region	Bucket Name	Specific Path	Operation
<input type="text" value="Select a region. v"/>	<input type="text" value="Enter the bucket nam"/>	<input type="text" value="/ Enter the resource  "/>	<input type="button" value="Delete"/>
<input type="button" value="Add Operation"/>			

Operation \*

Operation Name	Operation
Empty List	
<input type="button" value="Add Operation"/>	

Condition

Condition Key	Condition Oper...	Condition Value	Operation
Empty List			
<input type="button" value="Add Condition"/>			

**Result Feedback**

JSON Policy

```
{}
```

1. On the current page, select STS Policy as the policy type.
2. Parameter descriptions are provided below. For further details about configuration items, please refer to: [Obtaining Temporary Credentials for Federated Identities & Access Policy Language Overview](#)

Parameter	Required? (Yes or No)	Attribute	Description
Effect	Yes	Effect	Support selecting "Allow" or "Deny", corresponding to "Allow" and "Deny" in the policy syntax.
Resource Path	Yes	Resource	<p>The resource path consists of region, bucket name, and specific path, and supports adding either an entire bucket or a specified resource.</p> <ul style="list-style-type: none"> <li>● Entire bucket: If you want to define the resource scope as an entire bucket, only the region and bucket name need to be filled in.</li> <li>● Specified resource: If you want to limit the resource scope to a specific folder, the region, bucket name, and specific path must be provided</li> </ul> <div style="border: 1px solid #00aaff; padding: 5px; margin-top: 10px;"> <p><b>! Note:</b> For details such as Region, Bucket Name, and Specific Path, please refer to the descriptions of the corresponding parameters in the current table for guidance</p> </div>
Region	Yes	/	The region where the bucket is located, for example, ap-beijing. For more information about regions, please refer to <a href="#">Regions and Endpoints</a> .
Bucket Name	Yes	/	The bucket name consists of a custom name and an APPID, for example: examplebucket-1250000000. For more information about bucket naming and APPID, please refer to <a href="#">Bucket Naming Conventions</a> , <a href="#">APPID</a> , and <a href="#">Glossary of Object Storage API Terms</a> .
			<p>1.If no specific path is specified, the default value "*".</p> <p>2.To specify a path, fill in the field as follows:</p>

Specific Resource Path	Yes	/	<ul style="list-style-type: none"> <li>● To target all objects in the /folder/ directory: enter folder/*</li> <li>● To target a specific object /folder/exampleobject :enter folder/exampleobject</li> <li>● To target all objects with the .txt extension in the bucket: enter *.txt</li> </ul>
Operation	Yes	Action	add or remove operations that require authorization. For further details about operations, please refer to the <a href="#">Operations List</a> .
condition	No	Condition	When granting permissions, you can specify conditions such as restricting the user's access IP address. For more details about conditions, please refer to <a href="#">Condition Keys Description and Usage Examples</a> .

#### 4. Generate Policy

Click **Generate Policy**. The result will be displayed in JSON format in the feedback panel on the right. This tool supports generating policies multiple times and provides one-click copy and one-click clear functionality.

Signature Tool

Get Signature URL

Policy Generator Tool

Request Tool

When authorizing, it is recommended to strictly adhere to the Principle of Least Privilege. Limit user access to specific operations (such as read-only permissions) and define resource prefixes to prevent granting unnecessary privileges, which could result in unintended privilege escalation and pose security risks to data.

Policy type \*  CAM Policy  Bucket Policy  STS Policy

Effect \*  Allow  Deny

Region	Bucket Name	Specific Path	Ope...
ap-shanghai (Shangr	Enter the bucket nam	/ Enter the resource	Delete
<a href="#">Add Operation</a>			

Operation Name	Operation
Write Operations	Delete
<a href="#">Add Operation</a>	

Condition Key	Condition Operator	Condition Value	Operation
IP	Equal		Delete
<a href="#">Add Condition</a>			

Generate Policy

### Result Feedback

Clear All

JSON Policy

```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "allow",
      "action": [
        "name/cos:PutObject",
        "name/cos:PostObject",
        "name/cos:DeleteObject",
        "name/cos:InitiateMultipartUpload",
        "name/cos:UploadPart",
        "name/cos:CompleteMultipartUpload",
        "name/cos:AbortMultipartUpload"
      ],
      "resource": [
        "qcs::cos:ap-shanghai:uid/test:prefix//test/*"
      ],
      "condition": {
        "ip_equal": {
          "qcs:ip": [
            "198.162.1.1"
          ]
        }
      }
    }
  ]
}
```

# Self-Diagnosis Tool

Last updated: 2023-09-07 11:56:01

## Feature Overview

The Cloud Object Storage (COS) Self-Diagnostic Tool is a web-based tool provided by Tencent Cloud Object Storage to help users troubleshoot and diagnose erroneous requests. By entering the RequestId of the request (refer to [Obtaining RequestId Operation Guide](#) for instructions) on the tool's page and clicking "Diagnose," the tool will intelligently analyze the request and display its basic information, along with helpful guidance and diagnostic tips, assisting you in quickly identifying issues when using the COS API.

## Tool URL

Click here to access the [COS Self-Diagnostic Tool](#).

## How to Use

1. Click [Diagnosis Tool](#).
2. In the RequestId input box at the top, enter the RequestId to be diagnosed and click **Start Diagnosis**.
3. After a brief moment, you will see the corresponding intelligent diagnostic results. The diagnostic results are divided into two parts: Diagnosis Results and Request Information.
  - The suggestions help you quickly locate the COS API errors.
  - The request information is the information about the request corresponding to `RequestId`.
4. Provide Feedback on Diagnosis Results.

You can click **Helpful** or **Not Helpful** below the diagnosis results to share your opinion on the outcome with us, allowing us to further optimize the tool.

## FAQs

On the diagnostic tool page, you can also find the FAQs. If you have any queries, please [contact us](#).

## Supports and Limits

A COS `RequestId` must:

1. Start with N.
2. Contain at least 30 characters.

### 3. Comply with the Base64 standard.