# Cloud Object Storage

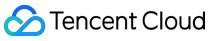
# ツールガイド

# 製品ドキュメント

**著作権声明**



**商標に関する声明**





**サービス声明**

本ドキュメントは、お客様にTencent Cloudの全部また一部の製品・サービスの概要をご紹介することを目的としておりますが、一部の製品・サービス内容は変更される可能性があります。お客様がご購入されるTencent Cloud製品・サービスの種類やサービス基準などは、お客様とTencent Cloudとの間の締結された商業契約に基づきます。別段の合意がない限り、Tencent Cloudは本ドキュメントの内容に関して、明示または黙示の一切保証もしません。

# カタログ:

# ツールガイド
# ツール概要

最終更新日：：2024-06-26 09:59:21

| ツール | 機能の説明 |
| --- | --- |
| COSBrowserツール | このツールは、ユーザーがビジュアルインターフェースを介して、データのアップロード、ダウンロードおよびアクセスリンクの発行といった操作を簡単に実行できるようにします。 |
| COSCLIツール | COSCLIは、Tencent CloudのCloud Object Storageが提供するクライアントサイドのコマンドラインツールです。COSCLIツールを使用すれば、シンプルなコマンドラインの指示で、COS内のオブジェクト(Object)に対し、一括アップロード・ダウンロード・削除などの操作を実行することができます。 |
| COSCMDツール | このツールは、ユーザーが簡単なコマンドラインを使用して、オブジェクトの一括アップロード・ダウンロード・削除などの操作を実行できるようサポートします。 |
| COS Migrationツール | このツールは、ユーザーのローカルや他のクラウドストレージなど、さまざまなソースデータアドレスからCOSへの移行をサポートします。 |
| FTP Serverツール | このツールは、ユーザーがFTPクライアントを使用してCOSから行うファイルのアップロード・ダウンロードをサポートします。 |
| COSFSツール | Linuxシステムでは、このツールを使用してバケットをローカルファイルシステムにマウントし、ローカルファイルシステム経由でCOS上のオブジェクトを操作することができます。 |
| Hadoopツール | Hadoopツールは、Hadoop、SparkおよびTezなどのビッグデータコンピューティングフレームワークをCOSに統合してサポートを提供することで、COSに保存されたデータの読み書きを容易に行うことができます。 |
| COSDistcpツール | COSDistcpは、MapReduceをベースとする分散ファイルコピーツールであり、主にHDFSとCOS間でデータをコピーするときに使います。 |
| Hadoop-cos-DistCheckerツール | Hadoop-cos-DistCheckerは、hadoop distcpコマンドを使用してHDFSからCOSにデータを移行した後、移行ディレクトリの整合性を検証するときに使います。 |
| HDFS TO COS | このツールは、HDFSからCOSにデータをコピーするときに使います。 |

| | |
|---|---|
| ツール | |
| 自己診断ツール | 自己診断ツールは、Tencent Cloud COSがユーザー向けに提供するWebツールで、エラーリクエストのセルフチェックとトラブルシューティングができます。 |

他のツールの要件がある場合は、ツール要件をお知らせください。速やかに要件の評価を行います。

# 環境のインストールと設定
# Javaのインストールと設定

最終更新日：：2024-06-26 09:59:21

JDKはJavaソフトウェア開発ツールキットです。ここでは、JDK 1.8バージョンを例として、WindowsとLinuxシステムにおけるJDKのインストールと環境設定の手順についてそれぞれご紹介します。

## Windows

### 1.JDKのダウンロード

Oracle公式サイト に進み、適切なJDKのバージョンをダウンロードします。

### 2. インストール

プロンプトに従って、ステップバイステップでインストールします。インストール中にインストールディレクトリをカスタマイズすることができます（デフォルトではCドライブにインストールします）。例えば、選択するインストールディレクトリは、`D:\Program Files\Java\jdk1.8.0_31` と `D:\Program Files\Java\jre1.8.0_31` です。

### 3. 設定

インストールが完了したら、**コンピュータ>プロパティ>高度なシステム設定>環境変数>システム変数>新規作成**を右クリックして、ソフトウェアをそれぞれ設定します。
変数名(N)：**JAVA_HOME**変数値(V)：`D:\Program Files\Java\jdk1.8.0_31`（実際のインストールパスに従って設定してください）。
変数名(N)：**CLASSPATH**変数値(V)：`.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;`（変数の値は `.` から始まりますので、ご注意ください）。
変数名(N)：**Path**
変数値(V)：`%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;`

### 4. テスト

設定が成功したかどうかをテストするには、**スタート（またはショートカットキー：Win+R）>実行（cmdを入力）>OK（またはEnterキーを押す）**をクリックし、コマンドjavacを入力してEnterキーを押します。下図のような情報が表示されれば、環境変数の設定は成功です。

## Linux

yumまたはapt-getコマンドを使用してopenjdkをインストールすると、クラスライブラリが不完全になり、ユーザーがインストール後に関連ツールを実行する際、エラーが報告される場合がありますので、ここでは手動で解凍してJDKをインストールする方法をお勧めします。具体的な手順は、次のとおりです。

### 1.JDKのダウンロード

Oracle公式サイト に進み、適切なJDKのバージョンをダウンロードし、インストールの準備を行います。

> ⚠ 注意：
> 以下は、例としてjdk-8u151-linux-x64.tar.gzを取り上げています。他のバージョンをダウンロードする場合は、ファイルの拡張子が.tar.gzであることに注意してください。

### 2. ディレクトリの作成

以下のコマンドを実行し、/usr/ディレクトリにjavaディレクトリを作成します。

```
mkdir /usr/java
cd /usr/java
```

ダウンロードしたファイルjdk-8u151-linux-x64.tar.gzを/usr/java/ディレクトリにコピーします。

### 3.JDKの解凍

次のコマンドを実行して、ファイルを解凍します。

```
tar -zxvf jdk-8u151-linux-x64.tar.gz
```

### 4. 環境変数の設定

/etc/profileファイルを編集し、profileファイルに次の内容を追加して保存します。

```
# set java environment
JAVA_HOME=/usr/java/jdk1.8.0_151
JRE_HOME=/usr/java/jdk1.8.0_151/jre
CLASS_PATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/li
b
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
export JAVA_HOME JRE_HOME CLASS_PATH PATH
```

> ⚠ 注意：
> ここでJAVA_HOME、JRE_HOMEは、実際のインストールパスとJDKのバージョンに応じて設定してください。

変更を有効にするには、以下を実行します。

```
source /etc/profile
```

### 5. テスト

次のコマンドを実行して、テストを行います。

```
java -version
```

Javaのバージョン情報が表示されれば、JDKのインストールは成功です。

```
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
```

# Pythonのインストールと設定

最終更新日：：2024-06-26 09:59:21

ここでは、さまざまなオペレーティングシステムにPython開発環境をインストールする方法を簡単にご紹介します。

## インストールパッケージによるインストール

### 1. ダウンロード

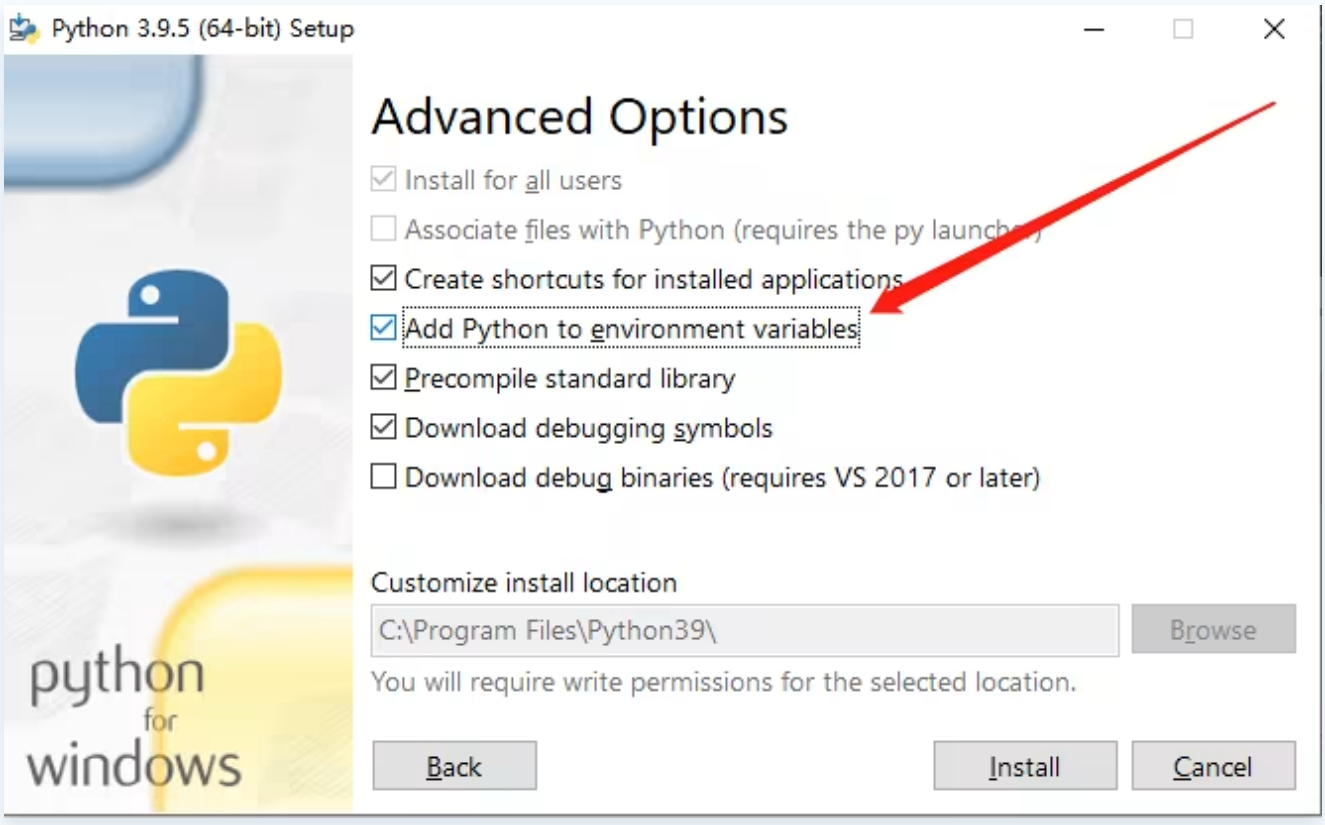
Python公式サイト にアクセスし、使用しているOSに応じて適切なインストールパッケージを選択し、ダウンロードします。

> **注意：**
> Pythonは、2020年1月1日をもってPython2のメンテナンスを終了しており、Python3のインストールが推奨されています。

### 2. インストール

インストールパッケージをダウンロードしたら、インストールパッケージの指示に従って、Python開発環境のインストールを完了します。

> **説明：**
> Windowsシステムユーザーは、インストール時にオプションの「Add Python to environment variables」にチェックを入れるようご注意ください。




### 3. 検証

端末で以下のコマンドを実行し、Pythonのバージョンを確認します。

```
python -V
```

端末からPythonのバージョン番号が出力されれば、インストールは成功です。

> **説明:**
> Windowsシステムユーザーはインストール完了後、コンピュータの再起動が必要になる場合があります。

### 4. 環境変数の設定

Windowsシステムで上記のコマンドを実行した際、端末に「内部コマンドでも外部コマンドでもありません」と表示された場合は、【コンピュータ】>【プロパティ】>【高度なシステム設定】>【環境変数】>【システム変数(S)】で「Path」を編集し、Pythonのインストールパスを追加してください。

## パッケージマネージャーによるインストール

### Mac OS

Mac OSをお使いのユーザーは、あらかじめHomeBrewをインストールした上で、HomeBrew経由でPythonをインストールすることができます。

```
brew install python
```

## Ubuntu

Ubuntuをお使いのユーザーは、Ubuntuに付属しているapt(Advanced Packaging Tool)パッケージマネージャーを使用すると、Pythonをインストールすることができます。

```
sudo apt-get install python
```

## CentOS

CentOSをお使いのユーザーは、CentOSに付属しているyum(Yellow dog Updater, Modified)パッケージマネージャーを使用すると、Pythonをインストールすることができます。

```
sudo yum install -y python
```

# Hadoopのインストールとテスト

最終更新日：：2025-01-24 13:05:48

Hadoopツールは、Hadoop-2.7.2以降のバージョンに依存して、基盤となるファイルストレージシステムとしてTencent Cloud COSを使用し、上位層のコンピューティングタスクを実行する機能を実装しています。Hadoopクラスターを起動するには、主にスタンドアロン、疑似分散、完全分散という3つの主なモードがあります。ここでは主に、Hadoop-2.7.4バージョンを例として、完全分散Hadoop環境をビルド、およびwordcountの簡単なテストをご紹介します。

## 環境の準備

- 複数のマシンを準備します。
- システムのインストールと設定は、CentOS公式サイト からダウンロードしてインストールできます。この記事では、CentOS 7.3.1611システムバージョンを使用します。
- Java環境をインストールします。操作の詳細については、Javaのインストールと設定をご参照ください。
- Hadoopの利用可能なパッケージ Apache Hadoop Releases Download をインストールします。

## ネットワークの設定

`ifconfig -a` を使用して各マシンのIPを確認し、pingコマンドを使用して相互にpingを送信できるかどうかを確認し、各マシンのIPを記録します。

## CentOSの設定

### ホスト名の設定

それぞれのマシンに対応するホスト名を設定します。例えば「master」、「slave*」などです。

```
hostnamectl set-hostname master
```

#### hostsの設定

```
vi /etc/hosts
```

コンテンツの編集：

```
202.xxx.xxx.xxx master
202.xxx.xxx.xxx slave1
202.xxx.xxx.xxx slave2
202.xxx.xxx.xxx slave3
```

```
# IPアドレスを実際のIPに置き換えます
```

### ファイアウォールの無効化

```
systemctl status firewalld.service  # ファイアウォールのステータスをチェックします
systemctl stop firewalld.service  # ファイアウォールを無効にします
systemctl disable firewalld.service  # ファイアウォールの起動を無効にします
```

### 時刻同期

```
yum install -y ntp  # ntpサービスをインストールします
ntpdate cn.pool.ntp.org  # ネットワークの時刻を同期させます
```

### JDKのインストールと設定

JDKインストールパッケージ（jdk-8u144-linux-x64.tar.gzなど）を `root` ルートディレクトリにアップロードします。

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

### 各ホスト間でのJDKのコピー

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

### 各ホストのJDK環境変数の設定

```
vi /etc/profile
```

コンテンツの編集：

```
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
```

ファイルを保存した後、/etc/profileを有効にするために、以下のコマンドを実行します:

```
source /etc/profile           # 設定ファイルを有効にする
java -version                 # javaバージョンを確認する
```

### SSHキーレスアクセスの設定

各ホストのSSHサービスステータスを個別にチェックします。

```
systemctl status sshd.service  //SSHサービスのステータスをチェックします
yum install openssh-server openssh-clients  //SSHサービスをインストールしま
す。すでにインストールされている場合は、この手順は不要です
systemctl start sshd.service  //SSHサービスを開始します。すでにインストールされ
ている場合は、この手順は不要です
```

各ホストで個別にキーを発行します

```
ssh-keygen -t rsa  //キーの発行
```

slave1の場合:

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

slave2の場合:

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

という感じで..
masterの場合:

```
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
```

# Hadoopのインストールと設定

## Hadoopのインストール

### Hadoopのインストール

hadoopインストールパッケージ（hadoop-2.7.4.tar.gzなど）を `root` ルートディレクトリにアップロードします。

```
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
```

`hadoop-2.7.4/etc/hadoop` ディレクトリに移動し、次の操作に進みます。

### Hadoopの設定

1. `hadoop-env.sh` ファイルを変更して、以下を追加します

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

SSHポートがデフォルトの22でない場合は、 `hadoop-env.sh` ファイルで以下のように変更します。

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

2. `yarn-env.sh` を変更します

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

3. `slaves` を変更します

設定内容:

```
削除:
```

```
localhost
追加:
slave1
slave2
slave3
```

4. `core-site.xml` を変更します

```
<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. `hdfs-site.xml` を変更します

```
<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. `mapred-site.xml` を変更します

```
<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>
```

7. `mapred-site.xml.template` をコピーして、`mapred-site.xml` という名前にします

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

8. `yarn-site.xml` を変更します

```
<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. ホスト間でHadoopを複製します

```
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. 各ホストのHadoop環境変数を設定します

設定ファイルを開きます。

```
vi /etc/profile
```

コンテンツの編集：

```
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
```

設定ファイルを有効にします。

```
source /etc/profile
```

## Hadoopの起動

1. namenodeのフォーマット

```
cd /usr/hadoop-2.7.4/sbin
```

```
hdfs namenode -format
```

2. 起動

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

3. プロセスのチェック

masterホストにResourceManager、SecondaryNameNode、NameNodeなどが含まれている場合は、次のような表示があれば起動に成功しています。

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

各slaveホストにDataNode、NodeManagerなどが含まれている場合は、次のような表示があれば起動に成功しています。

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

## wordcountの実行

Hadoopにはwordcountルーチンが付属されているため、直接呼び出すことができます。Hadoopを起動した後、以下のコマンドを使用すれば、HDFS内のファイルを操作することができます。

```
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
```

上図に示されている結果は、Hadoopのインストールが成功したことを示しています。

## 出力ディレクトリの確認

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

## 出力結果の確認

```
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
```

> **説明:**
> スタンドアロンモードと疑似分散モードの詳細な操作方法については、公式サイトの Hadoop入門 をご参照ください。

# COSBrowserツール
# COSBrowser概要

最終更新日：：2025-10-16 16:39:38

COSBrowserはTencent Cloud Object Storage（COS）がリリースした視覚化インターフェースツールです。より簡単なインタラクションの使用を可能にし、COSリソースの確認、転送および管理を手軽に実現できます。データ移行またはデータの一括アップロードを行う場合は、Migration Service Platform（MSP）をご利用ください。現在COSBrowserはデスクトップ端末およびモバイル端末向けにご提供しています。詳細については以下をご参照ください。

- デスクトップの使用説明
- モバイル端末の使用説明

## ダウンロードアドレス

| COSBrowserカテゴリー | サポートプラットフォーム | システム要件 | ダウンロードアドレス |
|---|---|---|---|
| デスクトップ | Windows | Windows 7 32/64ビット以上、Windows Server 2008 R2 64ビット以上 | Windows |
| | macOS | macOS 10.13以上 | macOS |
| | Linux | グラフィカルインターフェースとサポートが必要です AppImage 形式<br>注意：CentOSでクライアントを起動するには、/cosbrowser.AppImage --no-sandboxを実行する必要があります | Linux |
| モバイル端末 | Android | Android 4.4以上 | Android |
| | iOS | iOS 11以上 | iOS |
| Web版 | Web | Chrome/FireFox/Safari/IE10以上のブラウザ | Web |

## デスクトップ機能リスト

COSBrowserデスクトップはリソース管理に重点を置いており、ユーザーはCOSBrowserからデータを一括でアップロード・ダウンロードすることができます。

> ⚠ 注意：
> COSBrowserデスクトップは、システムによって設定されたプロキシを使用してインターネットへの接続を試みます。プロキシ設定が正常であることを確認するか、インターネットに接続できないプロキシ設定を無効にしてください。
> - Windowsユーザーは、OSの「インターネットオプション」で確認することができます。
> - macOSユーザーは、「ネットワーク環境設定」で確認することができます。
> - Linuxユーザーは、システム設定 > ネットワーク > ネットワークプロキシで確認することができます。

COSBrowserデスクトップは、以下の機能をサポートします。

| 機能 | 機能説明 |
| --- | --- |
| バケットの作成/削除 | バケットの作成と削除をサポートします |
| バケットの詳細情報の表示 | バケットの基本情報の表示をサポートします |
| 統計データの表示 | バケットの現在のストレージ容量とオブジェクト総数の表示をサポートします |
| 権限管理 | バケットやオブジェクトに関する権限の変更をサポートします |
| バージョン管理の設定 | バケットのバージョン管理の有効化と一時停止をサポートします |
| アクセスパスの追加 | アクセスパスの追加をサポートします |
| ファイル/フォルダのアップロード | ファイルまたはフォルダのバケットへの単一アップロード、一括アップロード、増分アップロードをサポートします<br>注意：<br>• 一括アップロードするファイルは10万個以内とします<br>• 中断からの再開対応（Android 非対応）<br>• データ移行またはデータの一括アップロードを行う場合は、Migration Service Platform（MSP）をご利用ください |
| ファイル/フォルダのダウンロード | ファイルまたはフォルダのローカルへの個別ダウンロード、一括ダウンロード、差分ダウンロードをサポートします<br>注意：<br>• 一括ダウンロードするファイルは10万個以内とします<br>• 中断からの再開対応（Android 非対応） |
| ファイル/フォルダの削 | バケット内のファイルやフォルダの個別削除、一括削除をサポートします |

| | |
|---|---|
| 除 | |
| ファイルの同期 | ローカルファイルのバケットへのリアルタイム同期をサポートします |
| ファイルのコピーと貼り付け | 1つのディレクトリから別のディレクトリへのファイルまたはフォルダへの個別コピー、一括コピーをサポートします |
| ファイルのリネーム | バケット内のファイルのリネームをサポートします |
| フォルダの新規作成 | バケット内でのフォルダの新規作成をサポートします |
| ファイルの詳細を表示 | バケット内のファイルの基本情報の表示をサポートします |
| ファイルリンクの発行 | 一時的な署名をリクエストすることで、制限時間のあるファイルへのアクセスリンクの発行をサポートします |
| ファイル/フォルダの共有 | ファイルとフォルダの共有と、共有の有効時間の設定をサポートします |
| ファイルリンクのエクスポート | ファイルリンクの一括エクスポートをサポートします |
| ファイルプレビュー | バケット内のメディアファイル（画像・ビデオ・オーディオ）のプレビューをサポートします |
| ファイル検索 | プレフィックス検索によるバケット内のファイル検索をサポートします |
| バケットの検索 | 作成済みのバケットの検索をサポートします |
| バージョンの履歴またはファイルフラグメントの表示 | • バージョン管理が有効になっているバケット内のファイルのバージョン履歴の表示をサポートします<br>• バケット内のファイルフラグメントの詳細の表示をサポートします |
| ファイル比較 | ローカルフォルダとバケット内のファイルの比較をサポートします |
| ビデオトランスコード | MPS機能を有効にしているバケット内のファイルのトランスコードをサポートします |
| 権限承認コードの生成 | 権限承認コードによってCOSBrowserクライアントに一時的にログインできます |
| 画像処理 | ズーム、トリミング、回転などの基本的な画像処理およびテキストウォーターマーク、画像ウォーターマーク、処理後の画像リンクの生成をサポートします |
| ネットワークプロキシの設定 | COSにアクセスするためのネットワークプロキシの設定をサポートします |
| ファイルのアップロー | ファイルのアップロード・ダウンロードの伝送同時実行数の設定をサポートし |

| | |
|---|---|
| ド/ダウンロード同時実行数の設定 | ます |
| アップロード/ダウンロードマルチパート同時実行数の設定 | ファイルのマルチパートアップロード・ダウンロードのパート数の設定をサポートします |
| アップロード/ダウンロード失敗時の再試行回数の設定 | ファイルのアップロードとダウンロードが失敗した場合の再試行回数の設定をサポートします |
| シングルスレッドアップロード/ダウンロード速度制限の設定 | シングルスレッドのアップロードとダウンロードの速度制限の設定をサポートします |
| アップロードの二次チェックの設定 | バケットにアップロードされたファイルの二次チェックをサポートします |
| ローカルログの表示 | COSBrowserへのユーザー操作記録のローカルログ形式による保存をサポートします |

## モバイル端末機能リスト

COSBrowserモバイル端末は、リソースの表示とモニタリングに重点を置いており、ユーザーはCOSのストレージ量やトラフィックなどのデータをいつでもどこでもモニタリングすることができます。COSBrowserモバイル端末でサポートされている機能については、モバイル端末機能リスト をご参照ください。

## ログの更新

- デスクトップ更新ログ： changelog 。
- モバイル端末更新ログ： changelog_mobile 。

## フィードバックと提案

COSBrowserの使用に関してご質問やご提案がございましたら、フィードバックをお寄せください。

- デスクトップのフィードバック： issues 。
- モバイル端末のフィードバック： issues_mobile 。

# デスクトップでの使い方

最終更新日：：2025-11-17 17:44:59

## ダウンロードとインストール

### ソフトウェアのダウンロード

| サポートプラットフォーム | システム要件 | ダウンロードアドレス |
| --- | --- | --- |
| Windows | Windows 7 32/64ビット以上、Windows Server 2008 R2 64ビット以上 | Windows |
| macOS | macOS 10.13以上 | macOS |
| Linux | グラフィカルインターフェースとAppImage形式をサポートしている必要があります | Linux |
| Web版 | Chrome/Firefox/Safari/IE10以上などのブラウザ | Web |

### ソフトウェアのインストール

ユーザーはプラットフォームに応じて、プログラムパッケージによるインストール、解凍してインストール、インストールせずブラウザで直接使用する方法のいずれかを選択できます。

> **注意：**
> CentOSでクライアントを起動するには、ターミナルで `./cosbrowser.AppImage --no-sandbox` を実行します。

## ログインソフトウェア

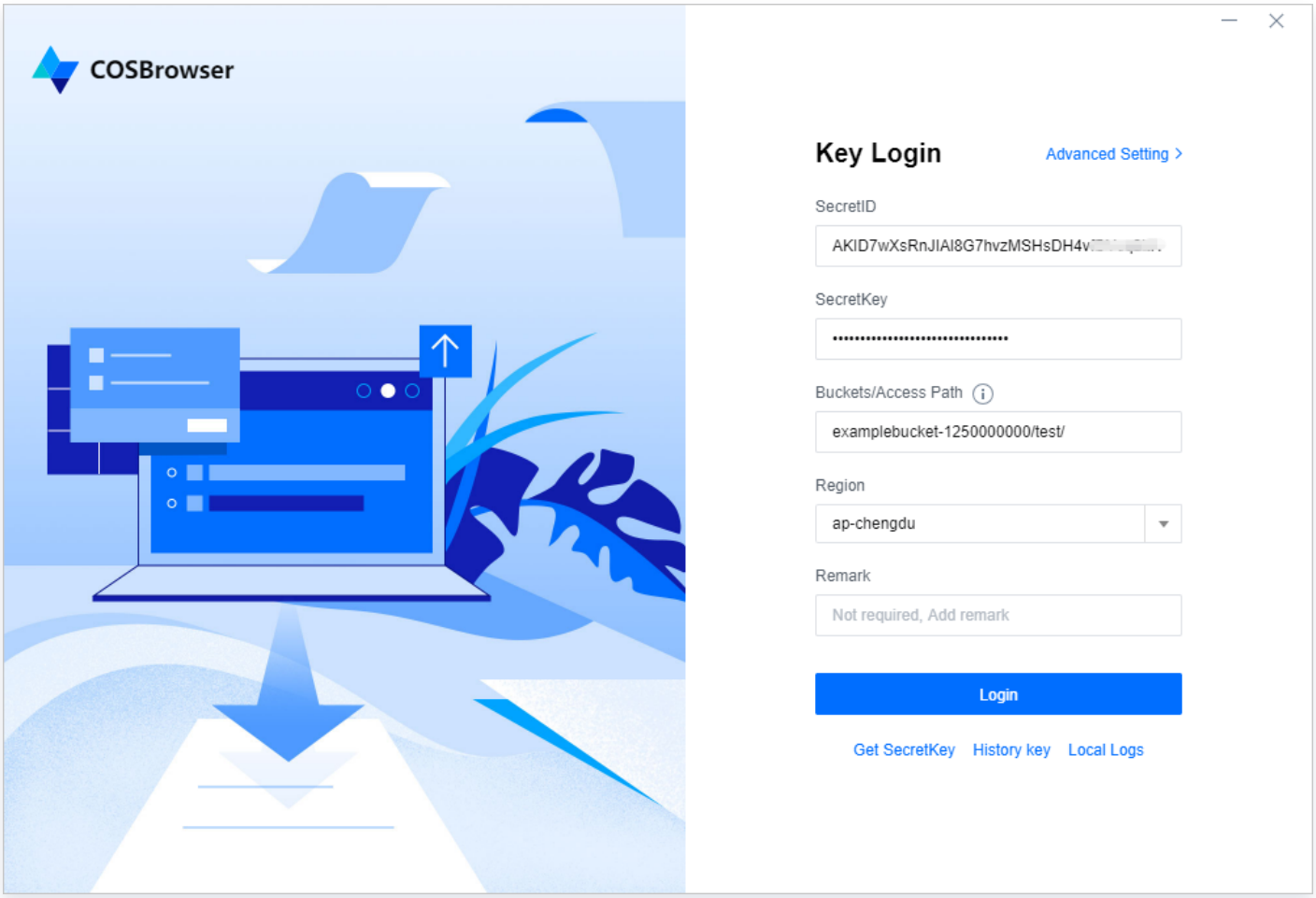
COSBrowserデスクトップは、パーマネントキーログイン、Tencent Cloudアカウントログイン、共有リンクログインという3つのログイン方法をサポートしています。同一アカウントでのマルチデバイス・マルチポイントログインもサポートしています。

### パーマネントキーログイン

Tencent Cloud APIキー（SecretID、SecretKey）でログインして使用します。このキーはCAMコンソールのAPIキー管理のページから作成、取得することができます。ログインに成功すると、キーは**履歴セッション**に保存され、次回から続けて使用できるようになります。ソフトウェアのログインインターフェースを以下に示します。その他の設定項目の説明は以下のとおりです。

- **バケット/アクセスパス**：現在使用しているキーがそのルートアカウントからアクセスを許可されている範囲が、バケットまたはバケット内のいずれかのディレクトリの権限のみである場合は、入力必須となります。入力後すぐに対応するファイルパスに入ることができます。形式はBucketまたはBucket/Object-prefixとなります。例えば、現在使用しているキーがバケットexamplebucket-1250000000内のdocフォルダへのアクセスのみ許可されている場合は、examplebucket-1250000000/docと入力します。
- **備考**：オペレーターや用途など、現在入力されているパーマネントキーの説明を記述することができます。履歴キーインターフェースで履歴セッションを管理する際、さまざまなSecretIDを区別することが可能です。
- **セッションの記憶**：
  - チェックが入っていない場合、現在のログインのみが有効となり、ログアウトすると入力されたTencent Cloud APIキーがクリアされます（現在のキーが履歴セッションに保存されている場合は、履歴セッションから削除されます）。
  - チェックが入っている場合、入力したTencent Cloud APIキーは記憶され、履歴セッションで管理できるようになります。

> ⚠ **注意：**
> COSBrowserは、プロジェクトキーでのログインをサポートしていません。




## Tencent Cloudアカウントでのログイン

**Tencent Cloudアカウントでのログイン**をクリックして、ポップアップウィンドウからTencent Cloudアカウントでの COSBrowserデスクトップにログインします。Tencent Cloudアカウントでサポートされているログイン方法には、WeChat、電子メール、QQ、WeChat公式アカウント、WeComおよびサブユーザーログインがあります。具体的なガイドについては、アカウント関連ドキュメント をご参照ください。

## 共有リンクログイン

他の人から転送または共有された**共有リンク**と**抽出コード**によって、COSBrowserデスクトップに一時的にログインすることができます。機能の説明については フォルダの共有 をご参照ください。

## 基本機能

> ⚠ 注意：
> この機能説明はWindowsのバージョンv 2.8.4の例です。その他のバージョンでは違いがありますので、更新ログ をご参照ください。

### 1. バケットの作成/削除

| 機能 | 説明 | 操作方法 |
| --- | --- | --- |
| バケットの作成 | クライアントから直接バケットを作成できます | 1. バケットリストで左上隅の**バケットの追加**をクリックします<br>2. バケット名を正しく入力し、リージョンやアクセス権限を選択します<br>3. **OK**をクリックすると、作成が完了します |
| バケットの削除 | バケットを削除する前に、バケット内のデータがクリアされていることを確認してください | 1. バケットリストで、対応するバケットの右側にある**削除**をクリックします<br>2. バケット内のデータがすべてクリアされていることを確認し、**OK**をクリックして削除します |

### 2. バケット詳細の表示

バケットリストの右側にある**詳細**アイコンをクリックすると、バケット詳細を表示できます。バケットの詳細情報には、バケット名、リージョン、アクセス権限、マルチAZ特性、バージョン管理、グローバルアクセラレーションなどのステータスおよびバケットドメイン名が含まれます。

> ⚠ 注意：
> マルチAZ特性は現在、北京、広州、上海、シンガポールリージョンなど、一部のリージョンでのみサポートされています。マルチAZ特性に関する説明は、マルチAZの特徴の概要 をご参照ください。

### 3. 統計データの表示

バケットリストの右側にある **...>統計** をクリックすると、バケットの統計データを表示できます。バケットデータには、ストレージの使用量とオブジェクト数が含まれます。

### 4. 権限管理

COSBrowserは、バケットとファイルの権限管理をサポートしています。

- バケット権限管理：バケットリストの右側にある操作バーの**権限管理**をクリックすると操作できます。

- オブジェクト権限管理：ファイルの右側にある操作バーの**権限管理**をクリックすると操作できます。

> **説明：**
> COSの権限に関する説明については、ACLの概要 をご参照ください。

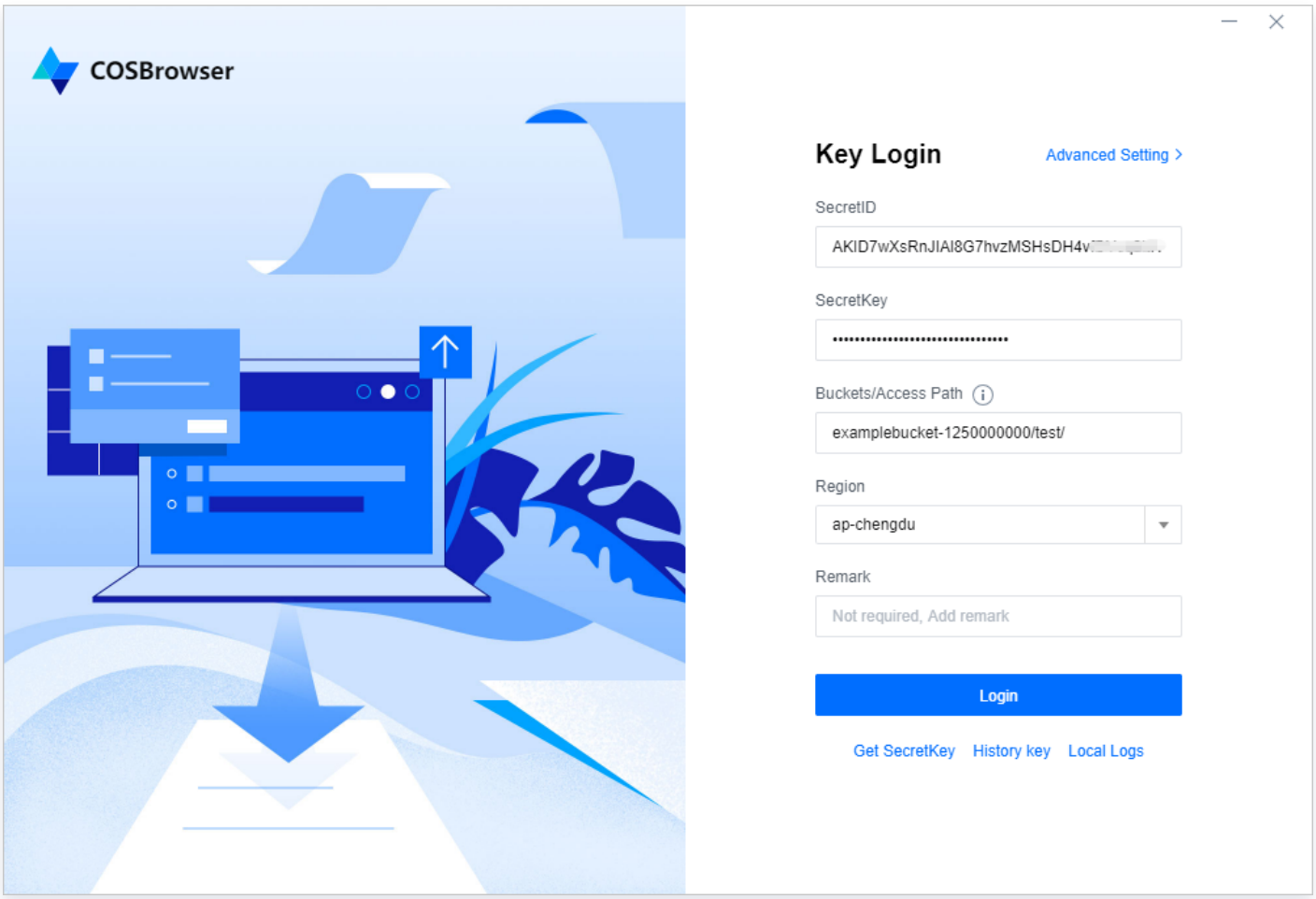
### 5. バージョン管理の設定

COSBrowserは、バケットのバージョン管理の有効化/無効化をサポートしています。
バケットリストの右側にある操作バーの**詳細**をクリックし、バケット詳細のポップアップウィンドウで、バージョン管理編集アイコンをクリックして操作できます。

> **説明：**
> バージョン管理に関する説明については、バージョン管理の概要 をご参照ください。

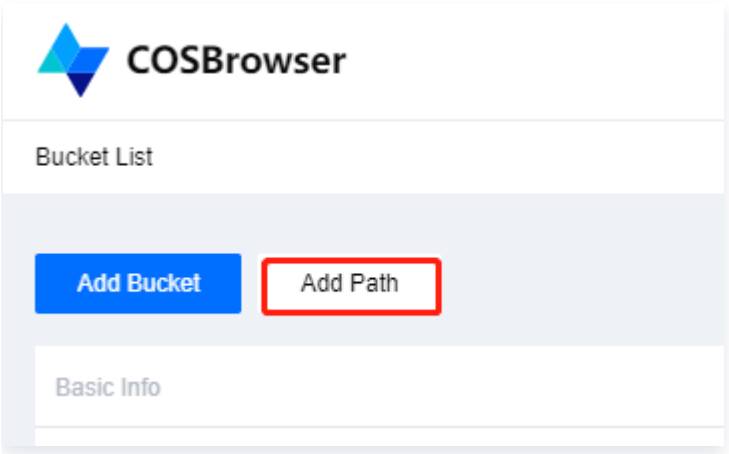
### 6. アクセスパスの追加

バケットリストにアクセスできないサブアカウントでログインしている場合、**アクセスパスを追加**することでアクセスできるようになります。COSBrowserでは、アクセスパスの追加方法として、次の2つの方法をご提供しています。
（1）ログインインターフェースに直接アクセスパスを追加し、対応するバケットのリージョン情報を選択します。ログインが完了すると、リソースを管理することができます。




（2）サブアカウントにログインした後、バケットリスト画面の左上隅にある**パスの追加**をクリックし、指定した

パスを入力してバケット管理リソースに進みます。




### 7. ファイル/フォルダのアップロード

| アップロード機能 | 説明 | 操作方法 |
| --- | --- | --- |
| ファイルのアップロード | COSBrowserは、単一または一括アップロードといった複数のアップロード方法をサポートしています | 指定したバケットまたはパスにファイルをアップロードするには、次のような方法があります。<br>1. アップロードをクリックしてファイルを選択し、ファイルを直接アップロードします。<br>2. ファイルリストの空白部分にあるファイルのアップロードを右クリックしてアップロードします。<br>3. ファイルをマウスでファイルリストウィンドウにドラッグしてアップロードします。 |
| フォルダとそのファイルのアップロード | バケットまたはパス内に同名のファイルやフォルダが存在する場合、デフォルトで上書きされます | 指定したバケットまたはパスにフォルダをアップロードするには、次のような方法があります。<br>1. アップロードをクリックしてフォルダを選択し、フォルダを直接アップロードします。<br>2. ファイルリストの空白部分にあるフォルダのアップロードを右クリックで選択してアップロードします。<br>3. フォルダをマウスでファイルリストウィンドウにドラッグすればアップロード完了です。 |
| インクリメンタルアップロード | インクリメンタルアップロードとは、アップロード操作を実行する前に、アップロードされたファイルとバケット内の既存のオブジェクトを比較し、同名のオブジェクトがある | インクリメンタルアップロード操作は、指定したバケットまたはパス内で、次のように実行します。<br>1. フォルダのアップロード方法を使用するには、「次へ」をクリックします。 |

| | | |
|---|---|---|
| | 場合、そのファイルをスキップしてアップロードしないことをいいます | 2. ストレージ方法でスキップを選択し、アップロードをクリックすると、インクリメンタルアップロードが完了します。 |

> ⚠ 注意：
> 大容量のファイルをアップロードする必要がある場合は、4コアでRAMが16GBのスペックのコンピュータをお勧めします（一度に最大30万ファイルのアップロードが可能です）。

## 8. ファイル/フォルダのダウンロード

| ダウンロード機能 | 説明 | 操作方法 |
|---|---|---|
| ファイルのダウンロード | COSBrowserは、単一または一括ダウンロードといった複数のダウンロード方法をサポートしています | ファイルをダウンロードするには、次のような方法があります。<br>1. ダウンロードしたいファイルを選択し、画面内のダウンロードをクリックすると、ファイルがダウンロードされます。<br>2. ファイルを選択し、ダウンロードを右クリックします。<br>3. ファイルをマウスでローカルにドラッグしてダウンロードします。 |
| フォルダとそのファイルのダウンロード | ローカルに同名のファイルやフォルダが存在する場合、デフォルトでリネームが行われます | フォルダとそのファイルをダウンロードするには、次のような方法があります。<br>1. ダウンロードしたいフォルダを選択し、画面内のダウンロードをクリックするとダウンロードできます。<br>2. ダウンロードを右クリックして、フォルダを直接ダウンロードします。<br>3. フォルダをマウスでローカルにドラッグしてダウンロードします。 |
| インクリメンタルダウンロード | インクリメンタルダウンロードとは、ダウンロード操作を実行する前に、ダウンロードされたオブジェクトとローカルファイルを比較し、同名のオブジェクトがある場合、そのオブジェクトをスキップしてダウンロードしないことをいいます | インクリメンタルダウンロード操作の手順は以下のとおりです。<br>1. ダウンロードしたいファイル/フォルダを選択し、マウスで右クリックします。<br>2. メニューの高度なダウンロードをクリックし、ポップアップウィンドウでスキップを選択します。<br>3. 次に、今すぐダウンロードをクリックすると、異なる名前のファイル/フォルダのインクリメンタルダウンロードが完了します。 |

> ⚠ 注意：

> 多数のファイルをダウンロードする必要がある場合は、4コアでRAMが16GBのスペックのコンピュータをお勧めします（一度に最大30万ファイルのダウンロードが可能です）。

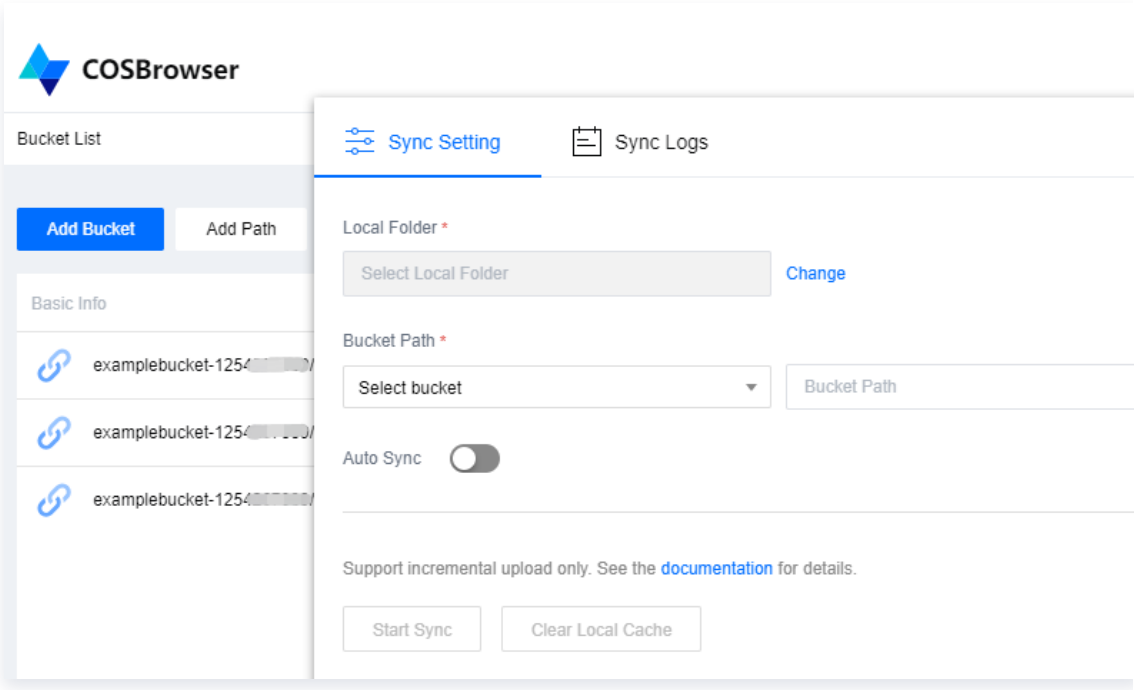
### 9. ファイル/フォルダの削除

削除したいファイル/フォルダを選択し、画面上の**その他**にある**削除**をクリックするか、**削除**を右クリックするとファイル/フォルダが削除されます。一括削除が可能です。

### 10. ファイル同期

ユーザーは、ファイル同期機能を使用することで、指定したローカルフォルダ内のファイルを自動でかつリアルタイムにバケット内にアップロードするか、または一定の時刻にバケット内にアップロードすることができます。具体的な操作手順は以下のとおりです。

1. 画面右上の**ツールボックス>ファイル同期**をクリックします。
2. ポップアップウィンドウで同期アップロードしたいローカルフォルダと、COSにアップロードするバケットディレクトリを指定します。
3. 同期タイプから1回のみの同期、自動同期、定時同期を選択できます。その後、**同期開始**をクリックすると、ファイル同期機能が有効になります。
4. ファイルの同期履歴は同期ログで確認できます。



> ⚠ 注意：
> - 同期とは、ファイルをアップロードする際に、バケットに同じファイルが存在するかどうかをシステムが自動認識し、同期機能によってバケットに存在しないファイルのみをアップロードすることをいいます。
> - 現在、ローカルファイルのストレージバケットへの同期アップロードのみサポートされており、逆方向の操作はサポートされていません。
> - ファイル同期機能は、手動同期、自動同期、定時同期の設定をサポートしています。

### 11. ファイルのコピー＆ペースト

指定したバケットまたはパス内で、コピーしたいファイル/フォルダを選択し、画面上の**その他**にある**コピー**をクリックするか、**コピー**を右クリックすると、ファイル/フォルダのコピーが完了します。コピーに成功すると、**その他のバケットまたはパス**に貼り付けることができます。一括コピー&ペーストに対応しています。

> ⚠ 注意：
> 貼り付け先のパスに、コピーしたファイルまたはフォルダと同名のファイルがある場合、デフォルトで上書きされます。

### 12. ファイルのリネーム

リネームしたいファイルを選択し、右クリックで**リネーム**を選択するか、ファイル右側の**その他の操作**の**リネーム**をクリックし、ファイル名を入力して確定すると、ファイルのリネームが完了します。

> ⓘ 説明：
> フォルダのリネーム操作は行えません。

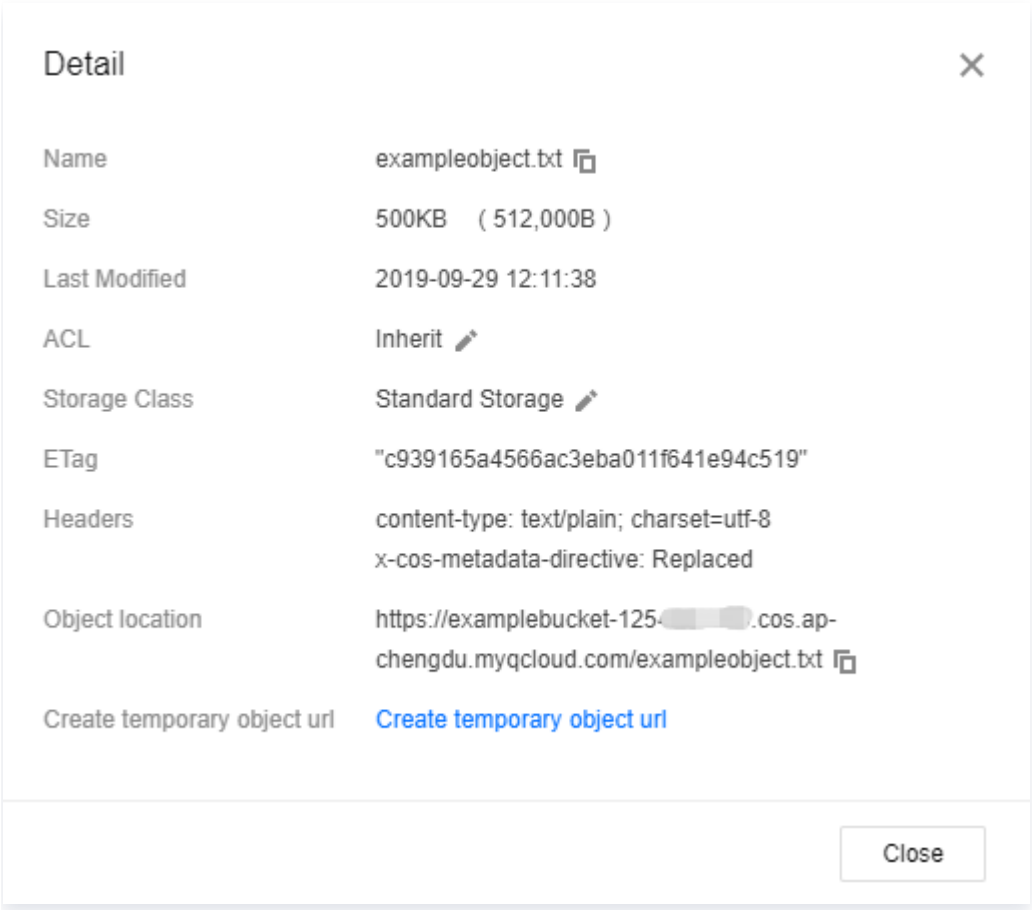
### 13. フォルダの新規作成

指定したバケットまたはパス内で、画面内の**フォルダの新規作成**をクリックするか、**フォルダの新規作成**を右クリックし、フォルダ名を入力して確定すると、フォルダの新規作成が完了します。

> ⚠ 注意：
> - フォルダ名は255文字までに制限されており、数字、英語およびこれらの文字を組み合わせて使用できます。
> - フォルダ名に、`\ / : * ? " | < >` などの特殊文字を含めることはできません。
> - `..` をフォルダ名にすることは許可されません。
> - フォルダのリネーム操作は行えませんので、慎重に命名してください。

### 14. ファイル詳細の表示

ファイル名をクリックするか、メニューの**詳細**を右クリックすると、ファイルの詳細を表示できます。ファイルの詳細情報には、ファイル名、ファイルサイズ、変更時刻、アクセス権限、ストレージタイプ、ETag、Headers、指定ドメイン名、オブジェクトアドレス、一時リンクの作成などが含まれます。




## 15. ファイルリンクの発行

COS内の各ファイルは、すべて特定のリンクを介してアクセスすることができます。ファイルにプライベート読み取り権限がある場合は、一時的な署名をリクエストすることで、時間制限付きの一時的なアクセスリンクを発行することができます。
ファイルリンクを発行するには、次のような方法があります。

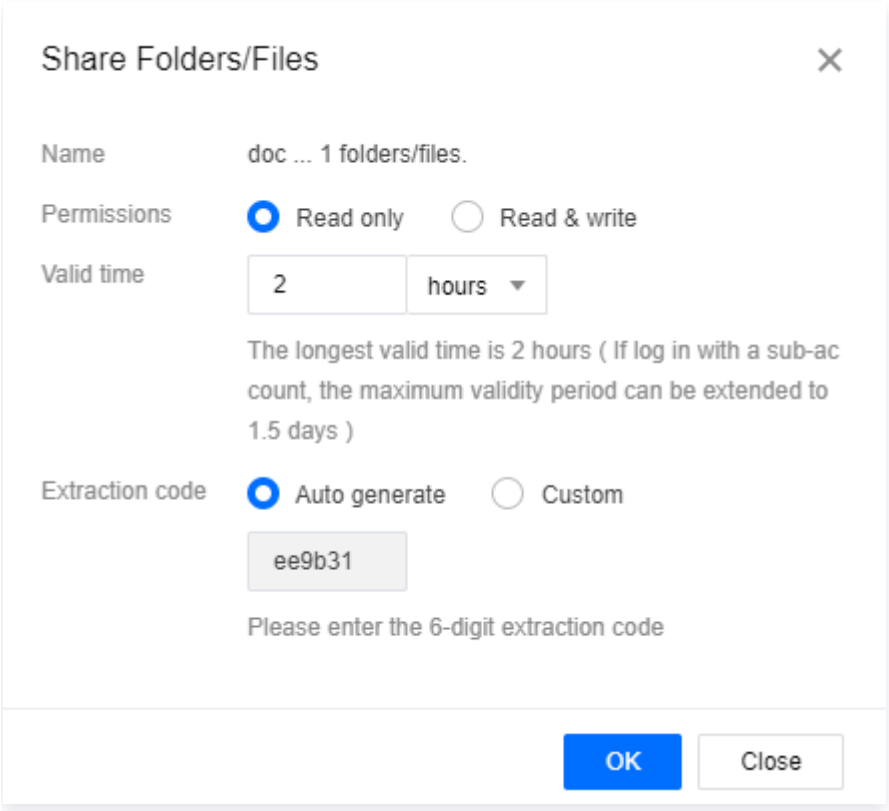
- テーブルビューで、ファイルの右側にある共有アイコンをクリックすると、ワンクリックでリンクが発行され、コピーすることができます。ファイルの権限がパブリック読み取り権限の場合、リンクは署名なしで永続的に有効になります。ファイルの権限がプライベート読み取り権限の場合、リンクは署名付きで2時間有効になります。
- ファイルを選択し、**リンクのコピー**を右クリックすると、ワンクリックでリンクが発行され、コピーすることができます。ファイルの権限がパブリック読み取り権限の場合、リンクは署名なしで永続的に有効になります。ファイルの権限がプライベート読み取り権限の場合、リンクは署名付きで2時間有効になります。
- ファイルの詳細で、**一時リンクの作成**をクリックすると、指定したドメイン名（CDNアクセラレーションドメイン名が有効になっている場合は設定可能など）の一時リンク、リンクタイプ、有効期間を設定することができます。

## 16. ファイル/フォルダの共有

操作バーの**共有**アイコンまたは右クリックメニューの**共有**オプションによって、COS内のいずれかのフォルダを共有でき、共有の有効期間を設定することも可能です。

**説明:**

- 単一フォルダの共有のみサポートされており、ファイルの一括共有はサポートされていません。
- 複数人でファイルを共有すると、ファイルのバージョンが混乱しやすくなります。バケットのバージョン管理機能を有効にして、ファイルのバージョン履歴を遡れるようにすることをお勧めします。




| パラメータ | 説明 |
| --- | --- |
| 権限 | 共有フォルダのアクセス権限を設定することができます。<br>• 読み取り専用: アクセスリンクからフォルダリストをプルし、フォルダ内のファイルをダウンロードすることができます。<br>• 読み書き アクセスリンクからフォルダリストのプル、フォルダからのファイルのダウンロード、共有フォルダへのファイルのアップロード、フォルダの新規作成ができます。 |
| 有効期間 | 単位は分、時間または日です。<br>• キーを使用してクライアントにログインする場合、ルートアカウントの有効期間の範囲は |

| | |
|---|---|
| | 1分～2時間、サブアカウントの有効期間の範囲は1分～1.5日です。<br>• Tencent Cloudアカウントを使用してクライアントにログインする場合、設定できる共有時間はすべて最大2時間までとなります。デフォルト値は、現在のアカウントで許可されている最大有効期間です。 |
| 抽出コード | 6桁で、デフォルトでシステムによって自動的に発行され、ユーザー定義が可能です。数字、アルファベット、記号の入力をサポートしています |

⚠ 注意：
リンクの有効期間中、共有リンクと抽出コードを受け取ったユーザーは、そのリンクを経由してフォルダにアクセスすることができます。

### 17. ファイルリンクのエクスポート

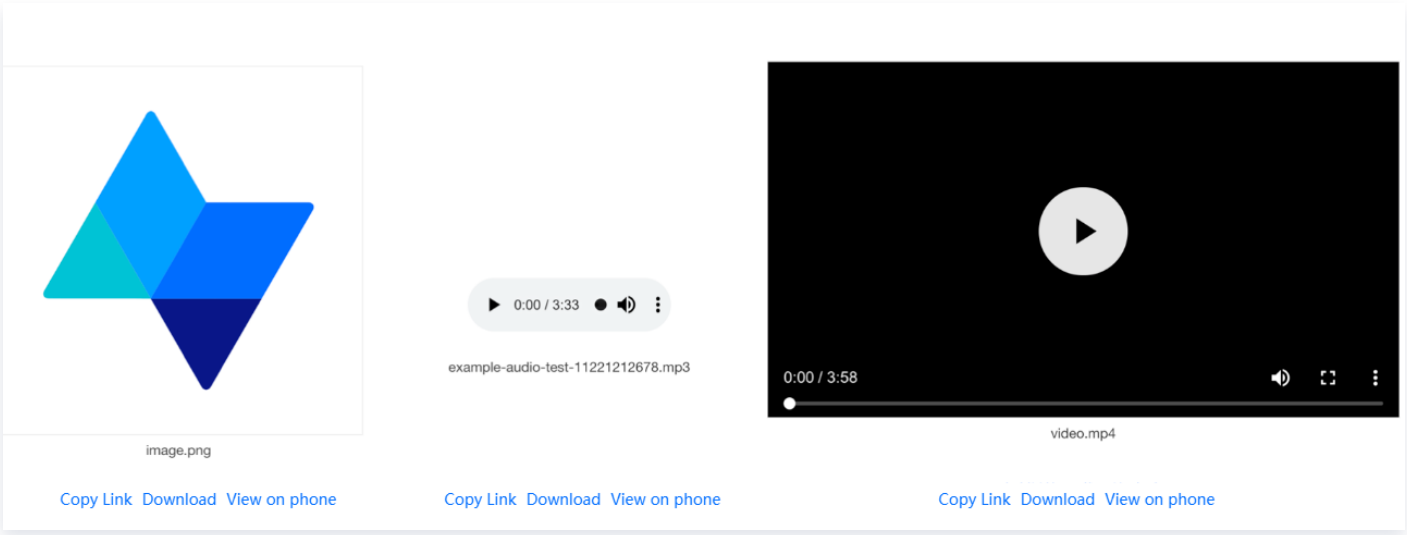
COSBrowserはファイルリンクのエクスポートをサポートしています。ツール画面右上のツールボックスのアイコンをクリックし、ツールボックスのポップアップウィンドウで**ファイルリンクのエクスポート**を選択します。その後、ファイルのあるバケット、エクスポートしたいフォルダパス（例えばルートパス下のfolderフォルダをエクスポートしたい場合は、`folder/` と入力します）および保存パスを入力し、最後に**エクスポート**をクリックすれば完了です。

### 18. フォルダのプレビュー

COSBrowserはメディア系ファイルのプレビューをサポートしており、現在は画像、ビデオ、オーディオに対応しています。メディア形式のファイルをダブルクリックするか、右クリックメニューの**プレビュー**または**再生**オプションをクリックすると、ファイルプレビュー画面を開くことができます。ファイルプレビューまたは再生画面では、次を選択することができます。

- **コピーリンク**：ファイルアクセスリンクを発行してコピーします。
- **ダウンロード**：ファイルをローカルにダウンロードします。ローカルに同名ファイルが存在する場合、デフォルトで上書きされます。
- **スマホでの表示**：プレビューインターフェースにファイルアクセスの2次元コードが発行されますので、スマホからこのコードを読み取れば、スマホから直接このファイルを表示することができます。

⚠ 注意：
- プレビューはほとんどの画像形式をサポートしています。ビデオ形式はmp4、webmのみ、オーディオ形式はmp3、wavのみをサポートしています。
- ファイルプレビューはダウンリンクのトラフィックを発生させるので、必要に応じて使用してください。




## 19. ファイル検索

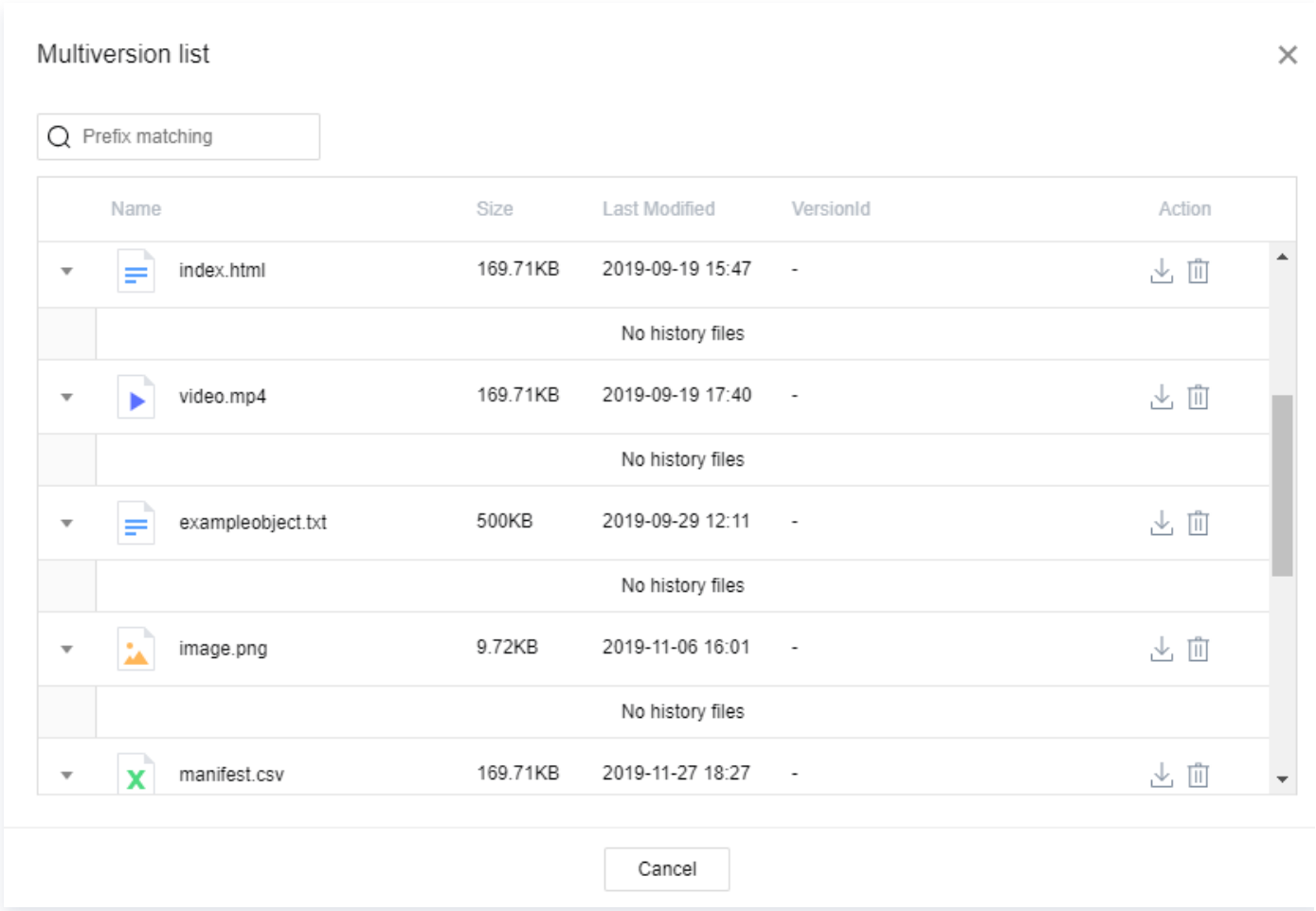
バケットの右上にある検索ボックスにファイル名を入力すると検索できます。COSBrowserは、ファイル名のプレフィックス検索とファイルあいまい検索をサポートしています。

## 20. バケットの検索

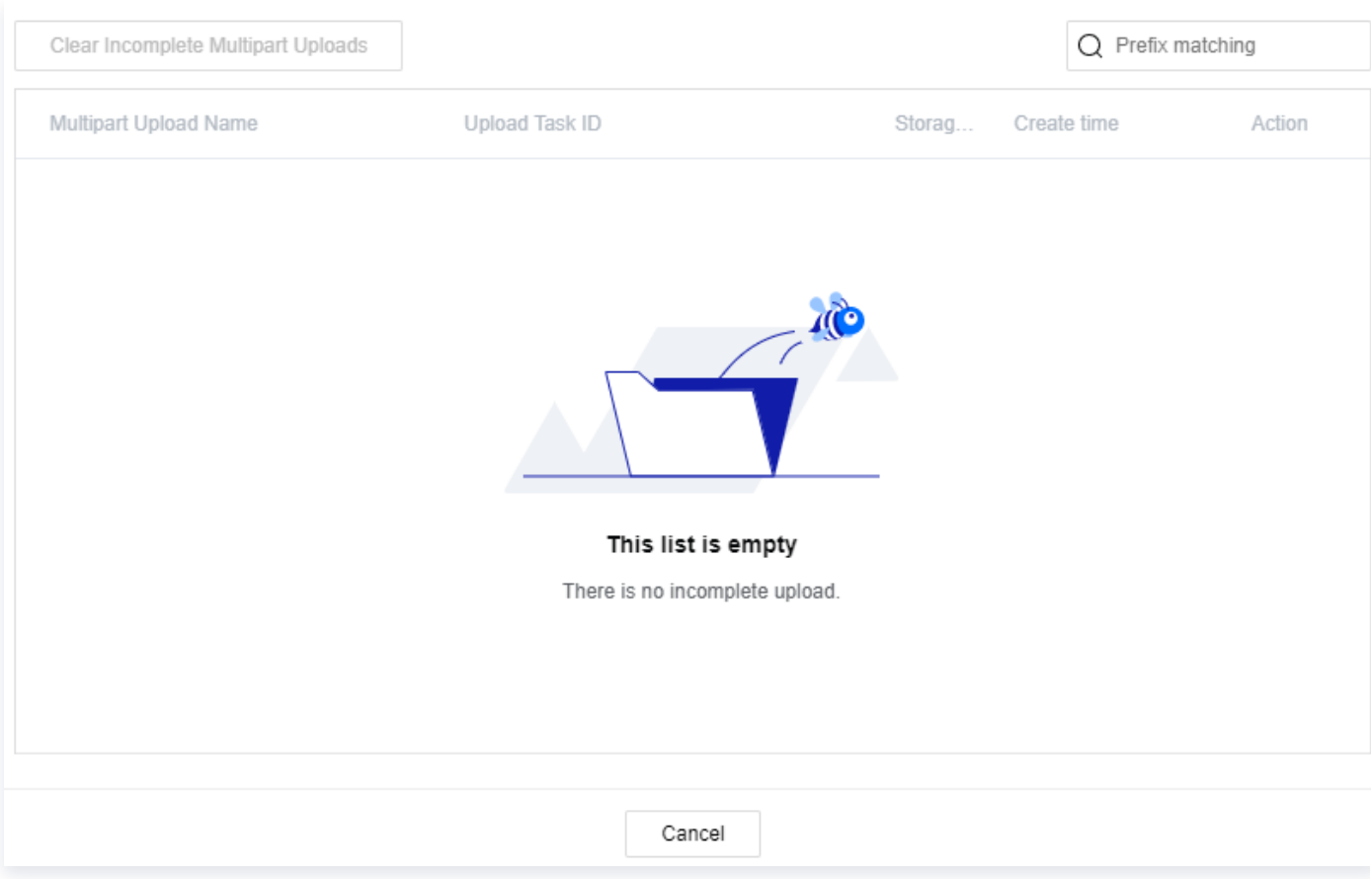
左側バケットリストの上にある検索ボックスにバケット名を入力すると、バケットをすばやく見つけることができます。

## 21. バージョン履歴またはファイルフラグメントの表示

- バケットのバージョン管理が有効になっている場合は、ファイルリストの上にある**表示>バージョン履歴**をクリックすると、ファイルのバージョン履歴を表示できます。バージョン履歴のリストでは、プレフィックス検索とすべてのバージョン履歴のクリア（最新バージョンのみ保持）をサポートしています。




- オブジェクトのアップロード中に、アップロードの一時停止やキャンセルを行うと、ファイルフラグメントが発生する場合があります。ファイルリストの上にある**表示>ファイルフラグメント**をクリックすると、ファイルフラグメントを表示できます。プレフィックス検索とすべてのファイルフラグメントのクリアに対応しています。




## 22. ファイルの比較

ツール画面右上のツールボックスのアイコンをクリックし、ツールボックスのポップアップウィンドウで**ファイルの比較**を選択します。その後、ローカルフォルダと、比較したいバケットを選択し、指定するリージョン、バケット、ディレクトリを選択し、最後に**比較を開始**をクリックします。

## 23. ビデオトランスコード

ツール画面右上のツールボックスのアイコンをクリックし、ツールボックスのポップアップウィンドウで**ビデオトランスコード**を選択します。その後、メディア処理サービスを有効化しているバケットを選択し、**トランスコードタスクの作成**をクリックします。トランスコードを行いたいメディアファイルとトランスコードテンプレートを選択し、トランスコード後のファイル名とストレージのパスを入力します。リージョン、バケット、ディレクトリを指定し、最後に**トランスコード**をクリックします。

## 24. 権限承認コードの生成

権限承認コードの生成機能によって、指定のバケット、バケット下のリソースおよび操作に対し、一時的な権限を承認することができます。フォルダ共有機能に比べてより柔軟に、カスタマイズした操作権限や特定のディレ

クトリに入る権限を付与することができます。一時的に使用可能なSecretId、SecretKey、Tokenおよび権限承認コードを生成することで、クライアントへの一時的なログインが可能になります。
操作に関する説明は次のとおりです。
ツール画面右上のツールボックスのアイコンをクリックし、ツールボックスのポップアップウィンドウで **権限承認コードの生成**を選択します。その後、権限承認コードを生成したいバケットと権限を承認するリソースの範囲を選択し、policy権限設定で権限を承認する操作（例えば読み取り書き込み権限など）を選択し、権限承認コードの有効期間を設定し、最後にOKをクリックします。

### 2. 画像処理

ツールの画像処理機能はズーム、トリミング、回転などの基本的な画像処理およびテキストウォーターマーク、画像ウォーターマーク、処理後の画像リンクの生成をサポートしています。
画像処理機能を使用したいバケットを選択した後、ツール画面右上のツールボックスのアイコンをクリックし、ツールボックスのポップアップウィンドウで**画像処理**を選択します。画像処理ポップアップウィンドウで、処理したい画像ファイルを選択し、機能オプションを設定した後、最後に**画像のプレビュー**をクリックすれば、処理後の画像リンクを生成できます。

## ソフトウェアの設定

| システム機能 | 説明 | 操作方法 |
| --- | --- | --- |
| ネットワークプロキシの設定 | COSBrowserはデフォルトで、システムによって設定されたプロキシを使用してネットワークへの接続を試みます。プロキシ設定が正常であることを確認するか、インターネットに接続できないプロキシ設定を無効にしてください | • 高度な設定 > プロキシを選択します。<br>• ネットワークプロキシを設定して、ネットワークに接続します。 |
| アップロード/ダウンロードファイル同時実行数の設定 | COSBrowserは、アップロード、ダウンロードファイル同時実行数の設定をサポートしています | • 高度な設定 > ダウンロード/アップロードを選択します。<br>• 一括転送の同時実行数を設定します。 |
| アップロード/ダウンロードマルチパート同 | COSBrowserは、マルチパートアップロードとマルチパートダウンロードをサポートしています。一定サイズ以上のファイル転送を行った場合、デフォルトでチャンクファイルによる転送になります | • 高度な設定 > ダウンロード/アップロードを選択します。<br>• マルチパート転送の同時実行数を設 |

| | | |
|---|---|---|
| 時実行数の設定 | | 定します。 |
| アップロード/ダウンロード失敗リトライ回数の設定 | COSBrowserは、ファイル転送時に失敗したタスクをデフォルトで再試行します | • 高度な設定 > ダウンロード/アップロードを選択します。<br>• 転送失敗のリトライ回数を設定します。 |
| シングルスレッドアップロード/ダウンロード速度制限の設定 | COSBrowserは、シングルスレッドのアップロード速度制限とシングルスレッドのダウンロード速度制限の設定をサポートしています。総アップロード（またはダウンロード）速度制限 = シングルスレッドアップロード（またはダウンロード）速度制限×ファイル同時実行数×マルチパート同時実行数 | • 高度な設定 > アップロード（またはダウンロード）を選択します。<br>• シングルスレッドのアップロード（またはダウンロード）速度制限を設定します。単位はMB/sです。 |
| アップロード時の2次チェックの設定 | COSBrowserは、アップロード後の2次チェックをサポートしており、オンラインファイルのサイズとステータスが正しいかどうかを確認します | • 高度な設定 > アップロードを選択します。<br>• アップロード完了後の2次チェックにチェックを入れます。 |
| ローカルログの表示 | COSBrowserはユーザーの操作を記録し、cosbrowser.logログの形式でローカルに保存します | • 高度な設定 > 関連を選択します。<br>• ローカルログをクリックすると、システムによってローカルログが配置されているディレクトリが開かれます。 |

# COSCLIツール
# COSCLI概要

最終更新日：：2025-05-19 15:52:06

COSCLIは、Tencent CloudのCloud Object Storage(COS)が提供するクライアントサイドのコマンドラインツールです。COSCLIツールを使用すれば、シンプルなコマンドラインの指示で、COS内のオブジェクト(Object)に対し、一括アップロード、ダウンロード、削除などの操作を実行することができます。
COSCLIは、Cobraフレームワークをベースにして記述されており、複数のバケットの設定やバケットをまたいだ操作をサポートしています。 `./coscli [command] --help` でCOSCLIの使用方法を確認することができます。

## 機能リスト

- 設定ファイルの発行と変更 – config
- バケットの作成 – mb
- バケットの削除 – rb
- ストレージバケットタグ－bucket－tagging
- バケットまたはファイルリストの照会 – ls
- さまざまなタイプのファイルの統計情報の取得 – du
- ファイルのアップロード・ダウンロード・コピー – cp
- ファイルの同時アップロード・ダウンロード・コピー – sync
- ファイルの削除 – rm
- ファイルハッシュ値の取得 – hash
- マルチパートアップロード中に発生したフラグメントのリストアップ – lsparts
- フラグメントのクリーンアップ – abort
- アーカイブファイルの取得 – restore
- 署名済みURLの取得 – signurl
- ソフトリンクの作成/取得 – symlink
- オブジェクト内容の確認 – cat
- ディレクトリ配下の内容のリストアップと統計 – lsdu
- バージョン管理 – bucket-versioning

## ダウンロード

COSCLIツールはWindows、MacOS、Linuxをサポートしています。詳細は ダウンロードとインストール をご参照ください。

# ダウンロードとインストール設定

最終更新日：：2025-11-12 16:22:47

COSCLIツールは、Windows、macOS、およびLinuxオペレーティングシステムのバイナリパッケージを提供しているため、簡単なインストールと設定後、使用可能です。

## 手順1：COSCLIツールをダウンロードする

操作シナリオに基づいてCOSCLIツールのダウンロードアドレスを選択してください。お使いのサーバーが中国にある場合は、中国国内サイトのダウンロードアドレスを使用することを推奨します（こちらのリンク先のツールバージョンはすべて最新版です。旧バージョンをご利用の場合は、 release にアクセスして履歴バージョンを取得できます）。

| 国内サイトのダウンロードアドレス | GitHubダウンロードアドレス(海外サイトでの使用を推奨) | SHA256 チェック |
| --- | --- | --- |
| Windows-386 | Windows-386 | 77c26d2e0226e16c1affa0e0c220cec2f3fd256461d0679162223d54336e5038 |
| Windows-amd64 | Windows-amd64 | 3bebe5d4dc0b82a0bc4e4e266ee9223b50aae3dd8c66fe3b54c9f1f2d03b4c97 |
| macOS-amd64 | macOS-amd64 | af76ceebe450d2c9656a221112ca0c64e6662545be7a6ad1a39ac888a978cd72 |
| macOS-arm64 | macOS-arm64 | 66b910ec6cc9d182c6b6625b3c6c7c9336fff1967198e5ca9b5581e6b6b8c0ac |
| Linux-386 | Linux-386 | 78801347f02aa4fac820fa5684b41a256e5c25f8fe9ed2be7a3c7dce90f892a4 |
| Linux-amd64 | Linux-amd64 | fc8d33b5b654b5a0d00c6f01019193c67c0855180b036cdb42cda18b735b736b |
| Linux-arm | Linux-arm | 54a6ee193a19a35b57179bcb8e9bb977063e400b76ba25c7f073fba4574ce553 |
| Linux-arm64 | Linux-arm64 | ca24967718c37e6961bf6167ea10b42b165d8a6fe3a7fa5f07225f0129461dbc |

また、国内サイトのmacOSとLinux環境におけるCOSCLIツールファイルをコマンドラインから取得することも可能です。

- macOS–amd64：
  `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-darwin-amd64`
- macOS–arm64：
  `wget https://cosbrowser.cloud.tencent.com/software/coscli/coscli-darwin-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-linux-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-linux-arm64`

**説明：**
GitHubでは、現在のバージョン番号はv1.0.7です。ツールの最新バージョン、過去のバージョンおとび更新ログを取得しようとする場合は、 release にアクセスして確認してください。

## 手順2：COSCLIツールをインストールする

### Windows

1. `Windows-amd64` を例として、ダウンロードされたWindowsバージョンのCOSCLIツールをC:\Users\<ユーザー名>ディレクトリに移動してください。
2. を `coscli-windows.exe` にリネームします `coscli.exe` 。
3. win+ r キーを押して実行プログラムを開きます。
4. ダイアログボックスで、 `cmd` を入力して `Enter` を押すと、コマンドラインウィンドウが表示されます。
5. コマンドプロンプトで、次のコマンドを入力します。

```
coscli --version
```

出力が `coscli version v1.0.7` であれば、インストールは成功です。

**説明：**
シ `Windows` ステムにおいて、異なるコマンドラインクライアントでCOSCLIを使用する方式がやや異なる可能性があります。 `coscli [command]` を入力した後にCOSCLIが正しく動作できない場合は、 `./coscli [command]` のフォーマットをお試しください。

## macOS

1. `macOS-amd64` を例として、次のコマンドを実行し、リネームmacOSバージョンのCOSCLIファイルは下記のとおりです。

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

2. 次のコマンドを実行してファイルの実行権限を修正します。

```
chmod 755 coscli
```

3. コマンドプロンプトで、次のコマンドを入力します。

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

出力が `coscli version v1.0.7` であれば、インストールは成功です。

> **説明:**
> macOSシステムにおいてCOSCLIを使用する場合、「開発者が確認できないため、coscliが開けられない」というプロンプトが表示された場合、設定〉安全性とプライバシー〉通用で「coscliを引き続き開く」を選択した後、COSCLIが正常に使用可能になります。

## Linux

1. `Linux-amd64` を例として、次のコマンドを実行し、 リネームLinuxバージョンのCOSCLIファイルは下記のとおりです。

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

2. 次のコマンドを実行してファイルの実行権限を修正します。

```
chmod 755 coscli
```

3. コマンドプロンプトで、次のコマンドを入力します。

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

出力が `coscli version v1.0.7` であれば、インストールは成功です。

## 手順3：COSCLIツールの構成

> **注意：**
> - 一時的な許可の使用方法によってツール使用の安全性がさらに向上できるため、ユーザーは 一時的キー でツールを使用することを推奨します。一時的キーを申請するときは、ターゲットストレージバケットやオブジェクト以外のリソースの漏洩を防止するために、 最小限の権限ガイドライン原則 に準拠してください。
> - 永久キーを使用する必要がある場合は、 最小限の権限ガイドライン原則 に準拠して永久キーの権限範囲を制限することを推奨します。

ユーザーはCOSCLIを初めて使用する場合、設定ファイルを初期化する必要があります。設定ファイルには次の2つの部分が含まれています。

- COSCLIにユーザーのTencent Cloudアカウントへのアクセスを許可するには、ユーザーはキーID、キーKey、一時的キーTokenを構成する必要があります。
- 通常バケットにエイリアスを追加するには、通常バケット名、バケットのリージョン情報、およびバケットエイリアスを構成する必要があります。通常バケットの情報を構成した後、ユーザーはバケット名やリージョン情報を入力せずに、エイリアスを使用してバケット操作を行うことが可能です。複数の通常バケット構成を追加した後、クロスバケットまたはクロスドメインの操作をより便利に行うことが可能です。通常バケット情報を構成する必要がない場合は、 `Enter` をクリックして次へ進みます。

初めて使用する場合、COSCLIが自動的に `./coscli config init` を呼び出して `~/.cos.yaml` で設定ファイルを生成するが、ユーザーはコマンドラインでインタラクティブに構成を完了することが可能です。その後、ユーザーは `./coscli config init` コマンドを使用して他の位置でCOSCLIのために設定ファイルをインタラクティブに生成することも可能です。ユーザーは `./coscli config show` を使用して設定ファイルの場所と構成パラメーター情報を確認することが可能です。
設定ファイルの中の各設定項目の説明は次のとおりです。

| 設定項目の | 必須入力 | 説明 |
| --- | --- | --- |
| Secret ID | はい | キーIDであり、使用上のリスクを軽減するために、サブアカウントキーを使用し、最小限の権限ガイドラインに準拠するよう許可することを推奨します。サブアカウントキーを取得するには、 サブアカウントのアクセスキーの管理 を参照してください。<br>**注意：**<br>初期化またはコマンドで設定したSecret IDは暗号化されて保存されます。設定ファイルを手動で変更する必要がある場合は、Disable Encryptionパラメータを設定してキーの暗号化を無効にする必要があります。 |

| | | |
|---|---|---|
| | | |
| Secret Key | はい | キーKeyであり、使用上のリスクを軽減するために、サブアカウントキーを使用し、最小限の権限ガイドラインに準拠するよう許可することを推奨します。サブアカウントキーを取得するには、 サブアカウントのアクセスキーの管理 を参照してください。<br><br>**注意：**<br>初期化またはコマンドで設定したSecret Keyは暗号化されて保存されます。設定ファイルを手動で変更する必要がある場合は、Disable Encryptionパラメータを設定してキーの暗号化を無効にする必要があります。 |
| Session Token | いいえ | 一時的キーtokenであり、一時キーを使用するときに設定する必要があるが、使用しない場合は、直接 `Enter` を押して次へ進みます。一時的キーの詳細については、 一時的キーを使用してCOS にアクセスする を参照してください。<br><br>**注意：**<br>初期化またはコマンドで設定したSession Tokenは暗号化されて保存されます。設定ファイルを手動で変更する必要がある場合は、Disable Encryptionパラメータを設定してキーの暗号化を無効にする必要があります。 |
| Mode | いいえ | はIDモードを設定し、列挙値の `SecretKey` と `CvmRole` をサポートします。空にすることが可能です。空である場合、値がデフォルトでは `SecretKey` になり、キーを使用してCOSをリクエストすることを示します。Modeが `CvmRole` になる場合、 インスタンスロールの管理 を使用してCOSをリクエストすることを示します。 |
| Cvm Role Name | いいえ | CVMロールのインスタンス名を設定します。 詳細については、 インスタンスロールの管理 を参照してください。 |
| protocol | いいえ | ネットワーク伝送プロトコルであり、デフォルトでは |

| | | |
|---|---|---|
| | | httpsです。Httpに変更する必要がある場合は、設定プロファイルに直接入って修正しても可能です。 |
| APPID | はい | APPIDは、ユーザーがTencent Cloudアカウントを申請した後に得られたアカウントであり、システムに自動的に割り当てられ、 アカウント情報 から取得できます。ストレージバケットのフルネームは `Bucket Name` と `APPID` という2つの要素からなり、フォーマットが `<BucketName-APPID>` です。詳細については、ストレージバケットの命名ルール を参照してください。 |
| Bucket Name | はい | ストレージバケット名であり、APPIDとともにストレージバケットのフルネームを構成します。フォーマットが `<BucketName-APPID>` です。詳細については、 ストレージバケットの命名ルール を参照してください。 |
| Bucket Endpoint | はい | ストレージバケットが存在するリージョンのドメイン名であり、ドメイン名のフォーマットがデフォルトでは `cos.<region>.myqcloud.com` です。その中、 `<region>` がストレージバケットのリージョンを表し、たとえば、ap-guangzhou、ap-beijingなどです。COSがサポートするリージョンリストは リージョンとアクセスドメイン名 を参照してください。<br>ストレージバケットではアクセラレーションが有効になっている場合は、グローバルアクセラレーションドメイン名を構成することことが可能です。たとえば、グローバルアクセラレーションドメイン名を `cos.accelerate.myqcloud.com` に構成し、イントラネットグローバルアクセラレーションドメイン名を `cos-internal.accelerate.tencentcos.cn` に構成します。 |
| Bucket Alias | いいえ | ストレージバケットのエイリアスであり、入力する必要があるコマンドの長さを減らすために、設定後に、使用時に `BucketName-APPID` の代わりに `BucketAlias` を使用することが可能です。この項目を構成しない場合は、 `BucketAlias` の値が `BucketName-APPID` の値になります。 |
| OFS Bucket | いいえ | メタデータアクセラレーションバケットのマークであり、該当のバケットでは メタデータアクセラレーション 機能が有効になっているかどうかを識別するために使用されます。 |
| CloseAutoSwitch | いいえ | バックアップドメインの自動切り替えを無効にするかど |

| | | |
|---|---|---|
| Host | | うかを設定します。値は true \| false を選択可能で、空にすることもできます。<br>• 設定されていないか、値が false の場合、バックアップドメインの切り替えが実行されます。<br>• true に設定された場合、バックアップドメインの切り替えは実行されません。 |
| DisableEncryption | いいえ | キー暗号化の無効化を設定します。選択可能の値はtrue \| false、空にすることもできます。<br>• 設定しないか、値がfalseの場合、設定ファイル内のキー関連情報が暗号化されます。<br>• true に設定すると、設定ファイル内のキー関連情報は暗号化されません。 |
| DisableAutoFetch BucketType | いいえ | バケットタイプの自動取得を無効にするかどうかを設定します。選択可能な値はtrue \| falseで、空にすることもできます。<br>• 設定しない、または値がfalseの場合、ツールは自動的にバケットタイプを取得します。これには `cos:HeadBucket` 権限が必要です。<br>• trueに設定した場合、ツールはバケットタイプを自動認識せず、まず `--bucket-type` に基づいてバケットタイプを決定します。このパラメータが設定されていない場合は、設定ファイル内のバケット情報のofsパラメータに基づいて決定します（trueはOFSバケット、falseはCOSバケット）。設定ファイルにバケット情報が含まれていない場合は、デフォルトでCOSバケットタイプが使用されます。 |

初回構成時には、COSCLIは1つのストレージバケットの構成のみを要求するが、複数のストレージバケットを構成しようとする場合は、後で `./coscli config add` コマンドを使用してストレージバケットの構成を追加することが可能です。設定ファイルを変更する必要がある場合、または設定ファイルに関する操作をさらに取得しようとする場合、 config コマンド を参照するか、または `./coscli config --help` コマンドを使用して設定ファイルに関連するコマンドを迅速に確認してください。
コマンドを正式に使用し開始する前に、 `./coscli --help` コマンドを使用してCOSCLIの使用方法を迅速に確認することが可能です。

## その他の構成方法

`./coscli config init` を使用して設定ファイルをインタラクティブに生成するほか、COSCLIの設定ファイルを直接手動で作成することも可能です。COSCLIの設定ファイルのフォーマットは `yaml` です。設定ファイルのフォーマットの例は次のとおりです。

```
cos:
  base:
    secretid: XXXXXXXXXXXXXXX
    secretkey: XXXXXXXXXXXXXXXXX
    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
```

**注意：**
COSCLI は、デフォルトでは~/.cos.yaml から構成項目を読み取ります。ユーザーがカスタム設定ファイルを使用しようとする場合は、コマンドの後に -c （--config-path）オプションを使用してください。設定ファイルに格納されている secretid/secretkey/sessiont secretid/secretkey/sessiontoken はすべて暗号化された文字列です。

# 一般オプション

最終更新日：：2025-11-12 16:22:47

「. /coscli --help」または「. /coscli -h」コマンドを使用して COSCLI がサポートする一般的なオプションを確認できます。

## オプションの説明

以下は、ツールのすべてのコマンドで使用できる COSCLI の一般的なオプションです。

**注意：**

- ユーザーがテンポラリクキー でールを利用し、一時的な権限付与により、ツール利用のセキュリティをさらに向上させることを推奨します。テンポラリクキーを申請する場合、最小権限ガイドライン に従い、ターゲットバケットまたはオブジェクトの外のリソースの漏洩を防止してください。
- パーマネントキーを使用しなければならない場合は、最小権限ガイドライン に従い、パーマネントキーの権限範囲を制限することを推奨します。

| オプション | 説明 |
| --- | --- |
| -h, --help | ヘルプ情報を出力するには、ユーザは -h または --help コマンドを使用することで、ツールのヘルプ情報や使用方法を確認することができます。また、各コマンドの後に（パラメータなし）-hを入力すると、そのコマンドの具体的な使い方を確認することができます。例えば、バケツ作成コマンドの具体的な使い方を確認するには、「coscli mb -h」と入力します。 |
| -c, --config-path | 構成ファイルのパス。COSCLI のデフォルトの構成ファイルのパスは「~/.cos.yaml」です。ユーザー定義の構成ファイルもサポートされています。コマンドの後に「-c」を使用して構成ファイルを指定できます。 |
| -e, --endpoint | 構成ファイルでバケットの地域を事前に設定しておくことに加え、COSCLI はコマンドの「-e」でバケットの endpoint を指定することもできます。 endpoint は「cos.<region>.myqcloud.com」のような形です。ここで、「<region>」はバケットの地域を表します。 例えば、「ap-guangzhou」、「ap-beijing」等です。COS がサポートしている地域のリストについては、地域とアクセスドメイン名 を参照してください。 |
| -i, --secret-id | COS へのアクセスに使用するキーの SecretId を指定します。 |
| -k, --secret-key | COS へのアクセスに使用するキーの SecretKey を指定します。 |
| --token | ユーザーが一テンポラリキーで COS にアクセスします。 |

| | |
|---|---|
| -v, --version | COSCLI のバージョンを表示します。 |
| -p, --protocol | ネットワーク伝送プロトコル。デフォルトは https です。 |
| --init-skip | デフォルトは false です。true(--init-skip=true)に設定すると、config init のインタラクティブ操作をスキップし、パラメータ内の SecretId、SecretKey、endpoint を直接使って api をリクエストします。このパラメータを使用すると、-i、-k、-e パラメータが必ず渡されます。 |
| --log-path | coscli.logファイルの場所をカスタマイズします。デフォルトではCOSCLIと同一のディレクトリです。フォルダまたは具体的なファイル（ファイル名は `.log` で終わる必要があります）を指定できます。例：/data/または/data/coscli.log |
| --customized | デフォルトではfalseです。trueに設定するとカスタムドメインが有効になり、--endpoint（-e）パラメータでカスタムドメインを指定する必要があります。 |
| --disable-log | デフォルトではfalseです。trueに設定するとcoscli.logの生成および対応するログの出力が無効になります。 |
| --bucket-type | 現在アクセスしているバケットのタイプ（COS/OFS）を指定します |

## 操作例

### 例1：バケットを切り替えてオブジェクトをアップロードします

ユーザが COSCLI で他の地域のバケットに切り替える必要がある場合、-e オプションでバケットの Endpoint を指定することができます。
例えば、ローカルのファイル test.txt を成都地域のバケット examplebucket-1250000000 にアップロードする場合、成都が読み込む endpoint は「cos.ap-chengdu.myqcloud.com」であり、コマンドは以下のようになります。

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

### 例2：ユーザーアカウントを切り替えてファイルリストを表示します

ユーザーが別のアカウントの ID を使用する必要がある場合は、-i と -k オプションでそれぞれユーザキーの SecretId と SecretKey を指定することもできます。
例えば、別アカウントの ID で成都地域配下のバケット examplebucket-1250000000 のファイルリストを表示する場合、コマンドは以下のようになります。

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

# よく使用するコマンド
# 設定ファイルの発行と変更 – config

最終更新日：：2025-11-14 15:30:16

## コマンド形式

以下のconfigコマンドは、設定ファイルを発行・変更するときに使います。

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

**説明：**

- 各設定項目を正しく入力すると、`./coscli config show`を使用して設定情報を確認することができます。
- このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。
- 生成された設定ファイルは、デフォルトで通信プロトコルがhttpsに設定されています。httpに変更する場合は、設定ファイルに直接アクセスして変更することができます。

configコマンドには、以下のサブコマンドが含まれます。

| command名 | commandの用途 |
| --- | --- |
| add | 新しいバケット設定を追加します。 |
| delete | 既存のバケット設定を削除します。 |
| init | 設定ファイルをインタラクティブに作成します。 |
| set | 設定ファイルのbaseグループの1つまたは複数の設定項目を変更します。baseグループには、`secretid`、`secretkey`、`sessiontoken`、`mode`、`cvmrolename`の情報が含まれています。 |
| show | 指定された設定ファイルの情報を印刷します。 |

configとそのサブコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| –h | ––help | ヘルプ情報を出力します。 |
| | | |

| | | |
|---|---|---|
| -c | --config-path | 使用する設定ファイルパスを指定します。 |

config addサブコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
|---|---|---|
| -h | --help | ヘルプ情報を出力します。 |
| -a | --alias | バケットエイリアス。 |
| -b | --bucket | バケット名。 |
| -r | --region | バケットリージョン。 |
| -o | --ofs | メタデータ加速バケトタグ。詳細については メタデータ加速概要 を参照してください。 |

> **注意：**
> 特定のストレージバケットのendpointを指定する場合は、一般flagの `-e` または `--endpoint` を使用できます。詳細については、一般オプションの紹介 をご参照ください。

config deleteサブコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
|---|---|---|
| -h | --help | ヘルプ情報を出力します。 |
| -a | --alias | バケットエイリアス。 |

config setサブコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
|---|---|---|
| -h | --help | ヘルプ情報を出力します。 |
| なし | --secret_id | secret IDは、コンソールへのアクセス から作成して取得することができます。 |
| なし | --secret_key | secret keyは、コンソールへのアクセス から作成して取得することができます。 |
| -t | --session_token | 一時的なキー token の設定。一時的なキーの詳細については、一時的なキーで COS にアクセスする を参照してください。 |
| なし | --mode | IDモードを設定します。列挙値 `SecretKey` と |

| | | |
|---|---|---|
| | | `CvmRole` をサポートしています。空にすることも可能で、その場合はデフォルトで `SecretKey` となり、キーを使用してCOSにリクエストを送信します。modeが `CvmRole` の場合、インスタンスロールの管理を使用してCOSにリクエストを送信します。 |
| なし | --cvm_role_name | CVMロールのインスタンス名を設定します。詳細については、インスタンスロールの管理をご参照ください。 |
| なし | --close_auto_switch_host | バックアップドメインへの自動切り替えを無効にするかどうかを設定します。選択可能な値はtrue、falseで、空にすることもできます。<br>• 設定しない、または値がfalseの場合、バックアップドメインへの切り替えが行われます。<br>• trueに設定した場合、バックアップドメインへの切り替えは行われません。 |
| なし | --disable_encryption | キーの暗号化を無効にするかどうかを設定します。デフォルトではfalseです。trueに設定した場合、設定ファイル内のキー関連情報は暗号化されません。 |
| なし | --disable_auto_fetch_bucket_type | バケットタイプの自動取得を無効にするかどうかを設定します。選択可能な値はtrue、falseです。<br>• 設定しない場合や、値がfalseの場合、バケットタイプが自動的に取得されます（使用するアカウントに `cos:HeadBucket` 権限があることを確認してください）。<br>• trueに設定した場合、バケットタイプは自動取得されず、ユーザーの設定ファイル内のバケットタイプが使用されます（例：設定ファイルのBucket設定項目ofsがtrueならバケットタイプはofs、falseならcosとなります）。 |

## 操作事例

### 新しいバケット設定の追加

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

### 既存のバケット設定の削除

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

### デフォルトの設定ファイルのsession-tokenの変更

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

### 指定された設定ファイルの情報の印刷

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

### デフォルト設定ファイルの mode と cvmrolename を変更します

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

# バケットの作成 – mb

最終更新日：：2025-11-14 15:30:16

mbコマンドは、バケットを作成するときに使います。

> **注意：**
> このコマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:PutBucket` に設定する必要があります。詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

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

> **注意：**
> mbコマンドを使用してストレージバケットを作成する際は、グローバルflagの `-e` または `--endpoint` を追加してストレージバケットのリージョンを指定する必要があります。

mbコマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<BucketName-APPID> | ストレージバケット名のカスタマイズに使用 | cos://examplebucket-1250000000 |

mbコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| -h | --help | ヘルプ情報を出力 |
| -r | --region | バケットリージョン |
| -m | --maz | マルチAZのストレージバケットを作成 |
| -o | --ofs | OFSのストレージバケットを作成 |
| なし | --acl | バケットのACLを設定します。例：private、public-read、public-read-write |
| なし | --grant-read | 権限を付与されるユーザーに、バケットの読み取り権限を与えます。形式はid="[OwnerUin]"、例： |

| | | |
|---|---|---|
| | | id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |
| なし | --grant-read-acp | 権限を付与されるユーザーに、バケットのアクセス制御リスト（ACL）の読み取り権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |
| なし | --grant-write-acp | 権限を付与されるユーザーに、バケットのアクセス制御リスト（ACL）の書き込み権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002" 。 |
| なし | --grant-full-control | 権限を付与されるユーザーに、バケットを操作するすべての権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例： id="100000000001",id="100000000002" 。 |
| なし | --tags | バケットタグのセット。最大50個のタグを設定できます（例：--tags="Key1=Value1&Key2=Value2"） |

**説明：**

- mbコマンドでストレージバケットを作成完了後、バケット別名でより簡単に操作できるためには、設定ファイルに当該ストレージバケットの情報を追加することを推奨します。具体的なコマンド使用方法については、以下の操作例をご参照ください。
- このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例：バケットbucket3-1250000000の作成

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

新規作成のストレージバケットに別名を設定する場合は、以下のコマンドを使用して設定ファイルを更新する必要があります。

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

cos://bucket3を更新すると、このバケットにアクセスできるようになります

# バケットの削除 – rb

最終更新日：：2025-05-19 15:52:07

rbコマンドは、バケットを削除するときに使います。

> **注意：**
> このコマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:DeleteBucket` に設定する必要があります。詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

```
./coscli rb cos://<BucketName-APPID> -r <Region> [flag]
```

rbコマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスするストレージバケットを指定します。パラメータ設定 内のバケット別名またはバケット名からアクセスできます。バケット名からアクセスする場合は、`endpoint` flagを追加する必要があります。 | バケット別名からアクセス：<br>cos://example-alias<br>バケット名からアクセス：<br>cos://examplebucket-1250000000 |

rbコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| -h | --help | ヘルプ情報を出力 |
| -r | --region | バケットリージョン |

> **说明：**
> - rbコマンドは空のストレージバケットのみを削除できます。ストレージバケット内にファイルがある場合は、ファイル削除 コマンドと ファイルチャンク削除 コマンドを使用して、ストレージバケット内のファイルとファイルチャンクをクリアしてからストレージバケットを削除してください。
> - rbコマンドを使用してストレージバケットを削除完了後、設定ファイルから当該ストレージバケットの情報を削除することを推奨します。
> - このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

```
// bucket3の削除（操作の確認が必要です）
./coscli rb cos://bucket3-1250000000 -e cos.ap-chengdu.myqcloud.com
```

```
// 構成ファイルを更新し、bucket3のストレージバケット構成を削除します
./coscli config delete -a bucket3
```

# ストレージバケットタグ－bucket－tagging

最終更新日：：2025-11-14 15:30:17

bucket-taggingコマンドは、ストレージバケットタグの作成（変更）取得削除に使用されます。各ストレージバケットごとに最大50組のタグがサポートされています。

**注意：**

- ストレージバケットタグを取得するには、ポリシー許可 を設定する際に、actionを `cos:GetBucketTagging` に設定する必要があります。
- ストレージバケットタグを設定するには、ポリシー許可 を設定する際に、actionを `cos:PutBucketTagging` に設定する必要があります。
- ストレージバケットタグを削除するには、ポリシー許可 を設定する際に、actionを `cos:DeleteBucketTagging` に設定する必要があります。

詳しい権限許可については、CAM対応API をご参照ください。

## コマンド形式

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

bucket-taggingコマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスするストレージバケットを指定します。パラメータ設定 内のバケット別名またはバケット名からアクセスできます。バケット名からアクセスする場合は、`endpoint` flagを追加する必要があります。 | バケット別名からアクセス：<br>cos://example-alias<br>バケット名からアクセス：<br>cos://examplebucket-1250000000 |

bucket-taggingコマンドには以下の選択可能なflagが含まれています：

| flag略語 | flag全称 | flag用途 |
| --- | --- | --- |
| -h | --help | コマンドの具体的な使用方法 |
| なし | --method | 実行する操作を指定します。put（タグ設定）、get（タグ照会）、delete（タグ削除）、add（タグ追加）が含まれます。 |

> **説明:**
> このコマンドに関するその他の一般オプション（ストレージバケット・ユーザーアカウントの切り替えなど）については、一般オプション をご参照ください。

## バケットタグの設定

バケットタグはキーバリューペア（Key-Value）で表されます。バケットの所有者およびPutBucketTagging権限を持つユーザーのみが設定でき、それ以外の場合はエラーコード `403 AccessDenied` が返されます。

### コマンド形式

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

`key#value` はキーバリューペアを示し、キーとバリューは#で区切られます。ストレージバケットにタグが設定されていない場合、このコマンドはストレージバケットに指定されたタグを追加します。ストレージバケットに既にタグが設定されている場合、このコマンドは既存のタグに上書きします。

### 操作例

バケットエイリアスがexample-aliasのバケットに2つのタグを設定します。1つ目のキーは1、バリューは111、2つ目のキーは2、バリューは222とします。コマンドは以下の通りです。

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

## ストレージバケットタグの検索

### コマンド形式

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

### 操作例

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

以下の出力結果は、バケット別名がexample-aliasのストレージバケットに2組のタグが設定されていることを示しています。その中で、1組はタグのキーが1、値が111で、もう1組はタグのキーが2、値が222です。

```
  KEY  |  VALUE
```

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

## バケットの全タグの削除

### コマンド形式

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

### 操作例

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

## バケットタグの追加

バケットタグはキーバリューペア（Key-Value）で表されます。バケットの所有者およびPutBucketTagging権限を持つユーザーのみが追加でき、それ以外の場合はエラーコード `403 AccessDenied` が返されます。

### コマンド形式

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

key#valueはキーバリューペアを示し、キーとバリューは#で区切られます。

### 操作例

バケットエイリアスがexample-aliasのバケットに1つのタグを追加します。キーは1、バリューは111とします。コマンドは以下の通りです。

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

# バケットまたはファイルリストの照会 – ls

最終更新日：：2025-05-19 15:52:07

lsコマンドは、すべてのバケットリスト、バケット内のファイルリストおよびフォルダ内のファイルリストを照会するために使います。

> ⚠ 注意：
> - ストレージバケット内のファイルを表示するには、ポリシー許可を設定する際に、actionを `cos:HeadBucket` 、 `cos:GetBucket` に設定する必要があります。
> - バージョン履歴情報を一覧表示するには（--all-versionsを渡す）、ポリシー許可を設定する際に、actionを `cos:HeadBucket` 、 `cos:GetBucketVersioning` 、 `cos:GetBucketObjectVersions` に設定する必要があります。
> - アカウントのストレージバケットを表示するには、ポリシー許可を設定する際に、actionを `cos:GetService` に設定する必要があります。
>
> 詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

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

lsコマンドには、次のオプションのパラメータが含まれます。

| パラメータ形式 | パラメータ用途 | 事例 |
| --- | --- | --- |
| cos://<bucket-name> | オプションのパラメータ。アクセスするバケットを指定します。パラメータ設定 のバケットの別名、またはバケット名を使用してアクセスすることがサポートされております。バケット名でアクセスする場合は、追加の `endpoint flag` が必要になります。 | バケットの別名を使用してアクセスする場合：`cos://example-alias`<br>バケット名を使用してアクセスする場合：`cos://examplebucket-1250000000` |
| /prefix/ | オプションのパラメータ。いずれかのフォルダを指定します | /picture/ |

lsコマンドには以下の選択可能なflagが含まれています：

| flag略語 | flag全称 | flag用途 |
| --- | --- | --- |
| -h | --help | コマンドの具体的な使用方法を確認 |

| | | |
|---|---|---|
| なし | --include | 特定モードのファイルを含む |
| なし | --exclude | 特定モードのファイルを除外する |
| -r | --recursive | フォルダを再帰的に走査し、すべてのファイルを一覧表示するかどうかを設定 |
| なし | --limit | リストアップの最大数を設定（0または未指定の場合はデフォルトで10000） |
| なし | --all-versions | オブジェクトのすべてのバージョンをリストします。これはバケットのバージョン管理を有効にした場合にのみ利用可能です。バージョン履歴のパラメータには、表示フィールド `VersionId` 、 `IsLatest` 、 `Delete Marker` が追加されます。 |

**説明：**

- `--include` と `--exclude` は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満たすファイルをフィルタリングすることができます。
- zshを使用する際、pattern文字列の両端に二重引用符を付ける必要がある場合があります。

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

- このコマンドに関するその他の一般オプション（ストレージバケット・ユーザーアカウントの切り替えなど）については、一般オプション をご参照ください。

## 操作事例

### 現在のアカウント下のすべてのバケットを一覧表示します

```
./coscli ls
```

返される情報には、バケット名、リージョン、作成時間、ストレージバケット総数が含まれます。例として：

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

```
                                TOTAL BUCKETS:   |          2
                               ------------------+---------------------
--
```

## ファイルの一覧表示

### bucket1にあるすべてのファイルを一覧表示します

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

返される情報には、オブジェクトキー（ストレージバケット内のオブジェクトの唯一の識別子）、ストレージタイプ、最終更新時間、オブジェクトサイズ、オブジェクト総数が含まれます。例として:

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

TOTAL OBJECTS:   |       1


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

### bucket1内のpictureフォルダにあるすべてのファイルとフォルダを一覧表示します

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

通常のリスト表示では、クエリパスのレベルのデータのみを返し、サブパスは展開しません。例として:

```
          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

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

**ストレージバケットbucket1のpictureフォルダー配下にあるすべてのファイルを再帰的にリストアップします**

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

クエリパスのレベルにサブパスがある場合、再帰的にリストアップするとすべてのサブパスがスキャンされ、クエリパスレベル以下のすべてのファイルが返されます。例として:

```


         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

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

**ストレージバケットbucket1内のすべての.mp4形式のファイルを再帰的にリストアップします**

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

**ストレージバケットbucket1内の.mp4形式以外のすべてのファイルを再帰的にリストアップします**

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

**ストレージバケットbucket1のpictureフォルダー配下にある、testで始まる、拡張子が.jpgではないすべてのファイルを再帰的にリストアップします**

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

## バージョン履歴を表示

### bucket1ストレージバケット内のすべてのバージョン履歴を表示

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

クエリパスのレベルにサブパスがある場合、再帰的にリストアップするとすべてのサブパスがスキャンされ、クエリパスレベル以下のすべてのファイルのすべてのバージョン履歴が返されます。例として:

```
               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 | MTg0NDUwMDM1MjIxMDM2MTY4MDc
| 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          |          | MTg0NDUwMDM1MjIwNjcxNzE5NDY
| true     | true          | 2025-02-26T14:34:02+08:00 |
|
  cmd/cmd/bucket_tagging.go      |          | MTg0NDUwMDM1MjIwNjcyNjczNzI
| 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

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

# さまざまなファイルの統計情報を取得 – du

最終更新日：：2025-05-19 15:52:08

duコマンドは、バケットまたはフォルダ内の各バケットタイプのファイルの統計情報を一覧表示するために使います。この統計情報には、異なるストレージタイプのファイルの総数や各タイプのファイルの合計サイズが含まれます。

**注意：**

- このコマンドですべてのオブジェクト情報を集計するには、ポリシー許可を設定する際に、actionを `cos:HeadBucket` 、 `cos:GetBucket` に設定する必要があります。
- このコマンドですべてのバージョン履歴情報を集計するには、ポリシー許可を設定する際に、actionを `cos:HeadBucket` 、 `cos:GetBucketVersioning` 、 `cos:GetBucketObjectVersions` に設定する必要があります。

詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

```
./coscli du cos://<bucketAlias>[/prefix/] [flag]
```

duコマンドには、次のオプションのパラメータが含まれます。

| パラメータ形式 | パラメータ用途 | 事例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスするバケットを指定します。パラメータ設定 のバケットの別名、またはバケット名を使用してアクセスすることがサポートされております。バケット名でアクセスする場合は、追加の `endpoint flag` が必要になります。 | バケットの別名を使用してアクセスする場合：`cos://example-alias` バケット名を使用してアクセスする場合：`cos://examplebucket-1250000000` |
| /prefix/ | いずれかのフォルダを指定します | cos://bucket1/picture/ |

duコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| –h | ––help | ヘルプ情報を出力 |
| なし | ––include | 特定のモードを含むファイル |

| | | |
|---|---|---|
| なし | --exclude | 特定のモードを除外したファイル |
| なし | --all-versions | オブジェクトのすべてのバージョンの統計。バケットのバージョン管理が有効になっている場合にのみ使用できます。統計には DeleteMarker の数も表示されます |

**説明:**

- `--include` と `--exclude` は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満たすファイルをフィルタリングすることができます。
- zshを使用する際、pattern文字列の両端に二重引用符を付ける必要がある場合があります。

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

- このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

### bucket1にあるファイルの統計情報を一覧表示します

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

返された結果の例は以下の通りで、出力情報には次のものが含まれます：そのバケット内の各ストレージタイプのオブジェクト数とサイズ、バケット内のオブジェクトの総数、バケット内のオブジェクトの総容量。

```
       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
```

### bucket1のpictureフォルダにあるファイルの統計情報を一覧表示します

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

### bucket1のpictureフォルダにあるすべての.mp4タイプのファイルの統計情報を一覧表示します

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

### bucket1のpictureフォルダにあるすべての.mdタイプのファイルの統計情報を一覧表示します

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

### bucket1ストレージバケット内のファイルおよびすべてのバージョン履歴の統計情報を一覧表示

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

返された結果の例は以下の通りで、出力情報には次のものが含まれます：そのバケット内の各ストレージタイプのオブジェクト数とサイズ、バケット内のオブジェクトの総数、バケット内のオブジェクトの総容量。

```
       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
```

# ファイルのアップロード・ダウンロードまたはコピー – cp

最終更新日：：2025-11-14 15:30:17

cpコマンドは、ファイルをアップロード、ダウンロードまたはコピーするときに使います。

> ⚠ 注意：
> - ファイルアップロードコマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:HeadBucket`、`cos:GetBucket`、`cos:HeadObject`、`cos:InitiateMultipartUpload`、`cos:UploadPart`、`cos:CompleteMultipartUpload`、`cos:ListMultipartUploads`、`cos:ListParts` に設定する必要があります。
> - ファイルダウンロードコマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:HeadBucket`、`cos:GetBucket`、`cos:HeadObject`、`cos:GetObject` に設定する必要があります。
> - ファイルコピーコマンドを使用するには、ポリシー許可を行う際に、ターゲットオブジェクトのアクションを `cos:GetBucket`、`cos:HeadObject`、`cos:InitiateMultipartUpload`、`cos:PutObject`、`cos:CompleteMultipartUpload` に、ソースオブジェクトのアクションは `cos:HeadBucket`、`cos:GetBucket`、`cos:HeadObject`、`cos:GetObject` に設定する必要があります。
>
> 詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

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

cp コマンドには以下のパラメータが含まれています。

| パラメータの形式 | パラメータの用途 | 例 |
| --- | --- | --- |
| source_path | ソースファイルのパス。ローカルパスまたは COS ファイルパスでもかまいません。COS パスは、構成パラメータ のバケットエイリアス、またはバケット名を使用してアクセスすることがサポートされています。 バケツ名でアクセスする場合は、さら | • ローカルパス：~/example.txt<br>• バケットエイリアスを使用して COS ファイルパスを指定する場合：cos://bucketalias/example.txt<br>• バケット名で COS ファイルパスを指定する場合：cos://examplebucket– |

| | | |
|---|---|---|
| | に「endpoint」flagを追加する必要があります。 | 1250000000/example.txt |
| destination_path | ターゲットファイルパス。ローカルパスまたは COS ファイルパスでもかまいません。COS パスは、構成パラメータ のバケットエイリアス、またはバケット名を使用してアクセスすることがサポートされています。 バケツ名でアクセスする場合は、さらに「endpoint」flagを追加する必要があります。 | • ローカルパス：~/example.txt<br>• バケットエイリアスを使用して COS ファイルパスを指定する場合：cos://bucketalias/example.txt<br>• バケット名で COS ファイルパスを指定する場合：cos://examplebucket-1250000000/example.txt |

cpコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
|---|---|---|
| なし | --include | 特定のモードを含むファイル（v1.0.4以前のバージョンではアップロード時にローカルファイル名のみをフィルタリングしていましたが、v1.0.4バージョン以降ではフルパスをフィルタリングします）<br>• えば、./test 以下のすべてのファイルをCOSにアップロードする必要があり、./test には aaa フォルダが含まれ、aaa フォルダ内には1.txt ファイルが含まれています。<br>○ 1.0.4より前のバージョンはaaa/1.txtにマッチします<br>○ 1.0.4以降のバージョンは./test/aaa/1.txtにマッチします |
| なし | --exclude | • 特定のモードを除外したファイル（v1.0.4以前のバージョンではアップロード時にローカルファイル名のみをフィルタリングしていましたが、v1.0.4バージョン以降ではフルパスをフィルタリングします）<br>• えば、./test 以下のすべてのファイルをCOSにアップロードする必要があり、./test には aaa フォルダが含まれ、aaa フォルダ内には1.txt ファイルが含まれています。<br>○ 1.0.4より前のバージョンはaaa/1.txtにマッチします<br>○ 1.0.4以降のバージョンは./test/aaa/1.txtにマッチします |
| -r | --recursive | フォルダ内のすべてのファイルを再帰的にトラバーサル処理するかどうか |
| なし | --storage-class | アップロードされるファイルのストレージタイプを指定します（デフォルトは STANDARD）。その他のストレージタイプについては、ストレージタイプの概要 を参照してください。 |
| なし | --part-size | ファイルチャンクサイズ（デフォルト32MB、最大5GBまで対応）。ファイルサイズに応じてチャンクサイズを自動調整する必要がある場合は、0に設定してください |
| | | |

| | | |
|---|---|---|
| なし | --thread-num | 同時実行スレッド数（デフォルトの同時実行は5） |
| なし | --rate-limiting | シングルリンクレート制限(0.1～100MB/s)、単位：MB/s |
| なし | --meta | アップロードファイルのメタ情報。一部のHTTP標準属性（HTTP Header）および `x-cos-meta-` で始まるユーザーカスタムメタデータ（User Meta）が含まれます。ファイルメタ情報の形式は `header:value#header:value` であり、例えば `Expires:2022-10-12T00:00:00.000Z#Cache-Control:no-cache#Content-Encoding:gzip#x-cos-meta-x:x` のようになります。 |
| なし | --routines | ファイル間の同時アップロードまたはダウンロードのスレッド数を指定します。デフォルトは「3」です。 |
| なし | --fail-output | このオプションは、アップロードまたはダウンロードに失敗したときにファイルのエラー出力を有効にするかどうかを決定します（デフォルトは「true」で、有効にする）。 有効にする場合、ファイル転送の失敗は指定したディレクトリに記録されます （指定しない場合、デフォルトで「./coscli_output’に記録される） 。 無効にする場合、エラーファイルの数のみがコンソールに出力されます。 |
| なし | --fail-output-path | このオプションは、アップロードまたはダウンロードに失敗したファイルが記録されるエラー出力フォルダを指定するために使用します。 カスタムフォルダパスを提供することで、エラー出力フォルダの場所と名前を管理できます。 このオプションが設定されていない場合、デフォルトのエラーログフォルダ「./coscli_output」が使用されま。 |
| なし | --retry-num | 頻度制限のリトライ回数 (デフォルトは「0」で、リトライしない)。 「1-10」回から選択可能。 複数のマシンが同じ COS ディレクトリに対して同時にダウンロード操作を行う場合、このパラメータを指定してリトライすることで、頻度制限エラーを回避することができます。 |
| なし | --err-retry-num | エラーのリトライ回数 (デフォルトは「0」)。「1-10」回指定するか、リトライしない場合は `0` を指定します。 |
| なし | --err-retry-interval | リトライ間隔 （「--err-retry-num」を「1-10」に指定した場合のみ有効)。リトライ間隔を「1-10」 秒で指定します。指定しない場合、または「0」に設定した場合は、各リトライ間隔は「1-10」秒以内でランダムになります。 |
| なし | --only-current-dir | 現在のディレクトリ内のファイルのみをアップロードし、サブディレクトリとその内容を無視するかどうか（デフォルトは「false」で、無視しない） |
| なし | --disable- | アップロード時にすべてのソフトリンクサブファイルとソフトリンクサブ |

| | | |
|---|---|---|
| | all-symlink | ディレクトリを無視するかどうか （デフォルトは「true」で、アップロードしない）。現在、LinuxおよびMacOSシステムのみサポートされています。 |
| なし | --enable-symlink-dir | ソフトリンクのサブディレクトリをアップロードするかどうか （デフォルトは「false」で、アップロードしない）。現在、LinuxおよびMacOSシステムのみサポートされています。 |
| なし | --disable-crc64 | CRC64 データチェックを無効にするかどうか。 （デフォルトは「false」で、チェックを有効にする） |
| なし | --disable-checksum | デフォルトはtrueで、シャードのCRC64のみを検証します。falseに設定すると、ファイル全体のCRC64を検証します。（coscli V1.0.6以前のバージョンではデフォルトはfalseです） |
| なし | --move | ファイルが目標パスにコピーされた後、ソースファイルを削除します。（COSパス間のみ使用可能） |
| なし | --version-id | 指定されたバージョンのファイルをダウンロードします。バージョン管理が有効になっているバケット内でのみ使用できます。（単一ファイルのみ対応） |
| なし | --process-log | プロセスログを有効にするかどうか。デフォルトはtrueで有効です |
| なし | --process-log-path | このオプションは、プロセスログを保存するための専用出力フォルダを指定するために使用します。ログには、エラーログ、正常実行ログ、リトライなどの詳細を含む、ファイルのアップロードまたはダウンロードに関する情報が記録されます。カスタムのフォルダパスを指定することで、ログ出力フォルダの場所と名前を制御できます。このオプションが設定されていない場合は、デフォルトのログフォルダ（coscli_output）が使用されます。 |
| なし | --skip-dir | デフォルトではfalseです。trueに設定すると、転送時にフォルダをスキップします。 |
| なし | --acl | ファイルのACLを設定します。例：private、public-read |
| なし | --grant-read | 権限を付与されるユーザーに、オブジェクトの読み取り権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002" 。 |
| なし | --grant-read-acp | 権限を付与されるユーザーに、オブジェクトのアクセス制御リスト（ACL）の読み取り権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002" 。 |
| なし | --grant-write-acp | 権限を付与されるユーザーに、オブジェクトのアクセス制御リスト（ACL）の書き込み権限を与えます。形式はid="[OwnerUin]"、例： |

| | | |
|---|---|---|
| | | id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002" 。 |
| なし | --grant-full-control | 権限を付与されるユーザーに、オブジェクトを操作するすべての権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002" 。 |
| なし | --tags | オブジェクトのタグセット。最大10個のタグを設定できます（例：--tags="Key1=Value1&Key2=Value2"） |
| なし | --forbid-overwrite | バージョニングが有効でないバケットに対し、アップロード時に同名オブジェクトの上書きを禁止するかどうかを指定します。<br>• falseを指定した場合、デフォルトで同名オブジェクトを上書きします。<br>• trueを指定した場合、同名オブジェクトの上書きを禁止します。<br>バケットのバージョニングが有効または一時停止中の状態である場合、x-cos-forbid-overwriteリクエストヘッダーの設定は無効となり、同名オブジェクトの上書きが許可されます。 |
| なし | --encryption-type | サーバーサイド暗号化の方式（SSE-COS/SSE-C）。 |
| なし | --server-side-encryption | サーバーサイド暗号化アルゴリズム。AES256、cos/kmsをサポートします。<br>SSE-COSまたはSSE-KMSを使用する場合、このフィールドは必須項目です。 |
| なし | --sse-customer-algo | サーバーサイド暗号化アルゴリズム。AES256をサポートします。<br>SSE-Cを使用する場合、このフィールドは必須項目です。 |
| なし | --sse-customer-key | サーバーサイド暗号化キーのBase64エンコード。<br>例：MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=。<br>SSE-Cを使用する場合、このフィールドは必須項目です。 |
| なし | --sse-customer-key-md5 | サーバーサイド暗号化キーのMD5ハッシュ値。Base64でエンコードされています。<br>例：U5L61r7jcwdNvT7frmUG8g==。<br>SSE-Cを使用する場合、このフィールドは必須項目です。 |
| なし | --check-point | レジュームアップロードを有効にするかどうか。デフォルトではtrueで有効です。 |

**説明：**

• cpコマンドは、大容量ファイルのアップロードやダウンロードを行う場合、アップロード/ダウン

ロードの同時実行を自動的に有効にします。

- ファイルが `--part-size` よりも大きい場合、COSCLIはまず `--part-size` に従ってファイルをチャンクにし、次に `--thread-num` 個のスレッドを使用してアップロード/ダウンロードのタスクを同時に実行します。
- 各スレッドは1つのリンクを維持します。各リンクに対して `--rate-limiting` パラメータを使用すると、シングルリンクのレート制限ができます。同時アップロード/ダウンロードが有効な場合、合計レートは、 `--thread-num * --rate-limiting` となります。
- ファイルをチャンクでアップロード/ダウンロードする場合、デフォルトで中断からの再開が有効になります。
- `--include` と `--exclude` は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満たすファイルをフィルタリングすることができます。
- zshを使用する場合、pattern文字列の両端に二重引用符を付ける必要がある場合があります。

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

- windows の cmd でコマンドを使用する場合、「——」文字(中国語の横棒)は cmd に貼り付けると自動的に「--」に変わってしまうので、手動で入力する必要があります。
- このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

### アップロード操作

#### 単一ファイルのアップロード

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

#### ローカルのtestフォルダ内のすべてのファイルとフォルダをbucket1バケットのexampleフォルダにアップロードします

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

ローカルのtestフォルダとそのサブフォルダ内のすべての.mp4形式のファイルをbucket1バケットのexampleフォルダにアップロードします

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

ローカルのtestフォルダとそのサブフォルダ内の.md形式以外のすべてのファイルをbucket1バケットのexampleフォルダにアップロードします

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

ローカルのtestフォルダとそのサブフォルダ内の.md形式および.html形式以外のすべてのファイルをbucket1バケットのexampleフォルダにアップロードします

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

ローカルのdirフォルダにはdirA、dirB、dirC、dirDという4つのフォルダがあり、dirフォルダにあるdirDフォルダを除くすべての内容をアップロードします

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

ローカルのtestフォルダ内のすべてのファイルとフォルダを、bucket1バケットのexampleフォルダにアップロードし、アーカイブ形式で保存します

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

ローカルのfile.txtファイルをバケットbucket1にアップロードし、シングルリンクレート制限を1.3MB/sに設定します

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

## ダウンロード操作

### 単一ファイルのダウンロード

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

bucket1バケットのexampleフォルダ内のすべてのファイルとフォルダをローカルのtestフォルダにダウンロードします

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

bucket1バケットのexampleフォルダとそのサブフォルダ内のすべての.mp4形式のファイルをローカルのtestフォルダにダウンロードしますードします

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

bucket1バケットのexampleフォルダとそのサブフォルダ内の.md形式以外のすべてのファイルをローカルのtestフォルダにダウンロードします

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

bucket1バケットのexampleフォルダとそのサブフォルダ内の.md形式および.html形式以外のすべてのファイルをローカルのtestフォルダにダウンロードします

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

bucket1バケットのexample.txtファイルのxxxバージョンをローカルのtestディレクトリにダウンロード

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

## コピー操作

### バケット内の単一ファイルのコピー

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

### バケット間の単一ファイルのコピー

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

bucket1バケットのexample1フォルダ内のすべてのファイルとフォルダをbucket2バケットのexample2フォルダにコピーします

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

bucket1バケットのexample1フォルダとそのサブフォルダー内のすべての.mp4形式のファイルをbucket2バケットのexample2フォルダにコピーします

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

bucket1バケットのexample1フォルダとそのサブフォルダ内の.md形式以外のすべてのファイルをbucket2バケットのexample2フォルダにコピーします

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

bucket1バケットのexample.txtファイルのxxxバージョンをbucket2バケットにコピー

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

bucket1のtestフォルダをbucket2に移動

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

# ファイルの同時アップロード・ダウンロードまたはコピー – sync

最終更新日：：2025-06-04 16:13:06

syncコマンドは、ファイルのアップロード・ダウンロード・コピーを同期させるときに使います。cpコマンドと異なる点は、syncコマンドはまず同名ファイルのcrc64を比較し、crc64の値が同じ場合は転送を行わない点です。

**注意：**

- ファイルアップロードコマンドを使用するには、ポリシー許可 を設定する際に、actionを `cos:HeadBucket` 、 `cos:GetBucket` 、 `cos:HeadObject` 、 `cos:InitiateMultipartUpload` 、 `cos:UploadPart` 、 `cos:CompleteMultipartUpload` 、 `cos:ListMultipartUploads` 、 `cos:ListParts` に設定する必要があります。
- ファイルダウンロードコマンドを使用するには、ポリシー許可 を設定する際に、actionを `cos:HeadBucket` 、 `cos:GetBucket` 、 `cos:HeadObject` 、 `cos:GetObject` に設定する必要があります。
- ファイルコピーコマンドを使用するには、ポリシー許可 を行う際に、ターゲットオブジェクトのアクションを `cos:GetBucket` 、 `cos:HeadObject` 、 `cos:InitiateMultipartUpload` 、 `cos:PutObject` 、 `cos:CompleteMultipartUpload` に、ソースオブジェクトのアクションは `cos:HeadBucket` 、 `cos:GetBucket` 、 `cos:HeadObject` 、 `cos:GetObject` に設定する必要があります。

詳細な権限については、CAM対応API をご参照ください。

## コマンド形式

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

sync コマンドには以下のパラメータが含まれています。

| パラメータの形式 | パラメータの用途 | 例 |
| --- | --- | --- |
| source_path | ソースファイルのパス。ローカルパスまたは COS ファイルパスでもかまいません。COS パスは、構成パラメータ のバケットエイリアス、またはバケット名を使用してアクセスすることがサポートさ | • ローカルパス：~/example.txt<br>• バケットエイリアスを使用して COS ファイルパスを指定する場合：cos://bucketalias/example.txt |

| | | |
|---|---|---|
| | れています。 バケツ名でアクセスする場合は、さらに「endpoint」flagを追加する必要があります。 | • バケット名で COS ファイルパスを指定する場合：cos://examplebucket-1250000000/example.txt |
| destination_path | ターゲットファイルパス。ローカルパスまたは COS ファイルパスでもかまいません。COS パスは、構成パラメータ のバケットエイリアス、またはバケット名を使用してアクセスすることがサポートされています。 バケツ名でアクセスする場合は、さらに「endpoint」flagを追加する必要があります。 | • ローカルパス：~/example.txt<br>• バケットエイリアスを使用して COS ファイルパスを指定する場合：cos://bucketalias/example.txt<br>• バケット名で COS ファイルパスを指定する場合：cos://examplebucket-1250000000/example.txt |

syncコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
|---|---|---|
| なし | --include | 特定のモードを含むファイル（v1.0.4以前のバージョンではアップロード時にローカルファイル名のみをフィルタリングしていましたが、v1.0.4バージョン以降ではフルパスをフィルタリングします） |
| なし | --exclude | 特定のモードを除外したファイル（v1.0.4以前のバージョンではアップロード時にローカルファイル名のみをフィルタリングしていましたが、v1.0.4バージョン以降ではフルパスをフィルタリングします） |
| -r | --recursive | フォルダ内のすべてのファイルを再帰的にトラバーサル処理するかどうか |
| なし | --storage-class | アップロードされるファイルのストレージタイプを指定します（デフォルトは STANDARD）。その他のストレージタイプについては、ストレージタイプの概要 を参照してください。 |
| なし | --part-size | ファイルチャンクサイズ（デフォルト32MB、最大5GBまで対応）。ファイルサイズに応じてチャンクサイズを自動調整する必要がある場合は、0に設定してください |
| なし | --thread-num | 同時実行スレッド数（デフォルトの同時実行は5） |
| なし | --rate-limiting | シングルリンクレート制限(0.1~100MB/s) |
| なし | --snapshot-path | ファイルのアップロードまたはダウンロード時のスナップショット情報が保存されるディレクトリを指定します。次にファイルがアップロードまたはダウンロードされる際に、coscli は指定されたディレクトリのスナップショット情報を読み込んで、増分的にアップロードまたはダウンロードを行いま |

| | | |
|---|---|---|
| | | す。このオプションは、ディレクトリファイルの同期を高速化するために使用されます。 |
| なし | --meta | アップロードされたファイルのメタ情報。これには HTTP 標準属性（HTTP Header）の一部と、`x-cos-meta-` で始まるユーザー定義メタデータ（User Meta）が含まれています。ファイルメタ情報の形式は `header:value#header:value` であり、その例は `Expires:2022-10-12T00:00:00.000Z#Cache-Control:no-cache#Content-Encoding:gzip#x-cos-meta-x:x` です。 |
| なし | --routines | ファイル間の並列アップロードまたはダウンロードスレッドのファイル数を指定します。デフォルトは `3` 。 |
| なし | --fail-output | になります。このオプションは、アップロードまたはダウンロードが失敗した時にファイルのエラー出力を有効にするかどうかを決定します（デフォルトは `true` で、有効になる）。有効にすると、ファイル転送の失敗が指定したディレクトリに記憶されます（指定しない場合はデフォルトで `./coscli_output` になる）。無効にすると、エラーファイルの数だけがコンソールに出力されます。 |
| なし | --fail-output-path | このオプションは、アップロードまたはダウンロードに失敗したファイルのエラー出力を記憶するフォルダを指定するために使用されます。カスタムフォルダパスを指定することで、エラー出力フォルダの場所と名前を管理できます。このオプションが設定されていない場合は、デフォルトのエラーログフォルダ `./coscli_output` が使用されます。 |
| なし | --retry-num | リトライ回数を制限します（デフォルトは `0` で、リトライしない）。 `1-10` 回から選択できます。複数のデバイスが同じ COS ディレクトリに対して同時にダウンロード操作を行う場合、このパラメータを指定してリトライすることで、頻度制限エラーを回避することができます。 |
| なし | --err-retry-num | エラーの場合のリトライ回数（デフォルトは `0` である）。 `1-10` 回を指定するか、リトライしない場合は `0` に設定します。 |
| なし | --err-retry-interval | リトライ間隔（ `--err-retry-num` が `1-10` に指定されている場合のみ利用可能）。リトライ間隔を `1-10` 秒で指定します。指定しないか、 `0` に設定した場合は、各リトライ間隔は `1-10` 秒の範囲でランダムになります。 |
| なし | --only-current-dir | カレントディレクトリ内のファイルのみをアップロードし、サブディレクトリとその内容を無視するかどうか (デフォルトは `false` で、無視しない)。 |
| なし | --disable-all-symlink | アップロード時にすべてのソフトリンクのサブファイルとソフトリンクのサブディレクトリを無視するかどうか (デフォルトは `true` で、アップロー |

| | | |
|---|---|---|
| | | ドしない)。現在、LinuxおよびMacOSシステムのみサポートされています。 |
| なし | --enable-symlink-dir | ソフトリンクのサブディレクトリをアップロードするかどうか (デフォルトは `false` で、アップロードしない)。現在、LinuxおよびMacOSシステムのみサポートされています。 |
| なし | --disable-crc64 | CRC64 データ検証を無効にするかどうか(デフォルトは `false` で、検証を有効にする）。 |
| なし | --delete | 指定されたターゲットパスにある他のファイルを削除し、今回同期されたファイルのみを残します（デ フォルトは `false` で、削除しない）。誤ってデータを削除しないように、--deleteオプションを使用する前にバージョン管理を有効にすることを推奨します。 |
| なし | --backup-dir | 削除されたファイルのバックアップを同期し、ターゲット側で削除され、ソース側には存在しないファイルの保存に使用されます（ダウンロードする時のみ有効で、 `--delete=true` の時に必ず伝送する）。アップロードとバケットコピーは、バージョン管理を使用して誤って削除されたデータを復元してください。 |
| なし | --force | 確認のプロンプトを出さずに操作を強制します（デフォルトは `false` ）。 |
| なし | --disable-checksum | デフォルトは false で、ファイル全体の crc64 をチェックします。true の場合はシャード crc64 のみをチェックします。 |

**説明:**

- syncコマンドは、大容量ファイルのアップロードやダウンロードを行う場合、アップロード/ダウンロードの同時実行を自動的に有効にします。
- ファイルが `--part-size` よりも大きい場合、COSCLIはまず `--part-size` に従ってファイルをチャンクにし、次に `--thread-num` 個のスレッドを使用してアップロード/ダウンロードのタスクを同時に実行します。
- 各スレッドは1つのリンクを維持します。各リンクに対して `--rate-limiting` パラメータを使用すると、シングルリンクのレート制限ができます。同時アップロード/ダウンロードが有効な場合、合計レートは、 `--thread-num * --rate-limiting` となります。
- ファイルをチャンクでアップロード/ダウンロードする場合、デフォルトで中断からの再開が有効になります。
- `--include` と `--exclude` は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満たすファイルをフィルタリングすることができます。
- zshを使用する場合、pattern文字列の両端に二重引用符を付ける必要がある場合があります。
- napshot-path を、移動するディレクトリまたはそのサブディレクトリに設定しないでください。

```
./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
```

- sync コマンドを使用する際に PUT リクエスト料金が発生することに加え、以下の 2 つのケースではクラウド上のファイルに対する HEAD リクエスト料金が発生し、追加の料金が発生することがあります。
  - スナップショットディレクトリを指定するためのパラメータ（`–snapshot–path``）が追加されていない場合、HEAD リクエスト料金が発生します。
  - 指定されたスナップショットディレクトリパラメータ（「–-snapshot–path」）が追加され、スナップショットディレクトリが初めて生成される場合、HEAD リクエスト料金が発生します。初めて生成されるスナップショットディレクトリでない場合は追加のリクエスト料金は発生しません。
- このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

### ファイルの同時アップロード

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

### ファイルの同時ダウンロード

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

### バケット内でのファイル同期コピー

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

### バケット間でのファイル同期コピー

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

# ファイルの削除 – rm

最終更新日：：2025-05-19 15:52:08

rmコマンドは、ファイルを削除するときに使います。

**注意：**

- rm コマンドを使用するには、COSCLI V1.0.1 以上のバージョンをダウンロードしてください。詳細は ダウンロードとインストールの設定 を参照してください。
- 現在お使いの COSCLI のバージョンが V1.0.0 の場合は、必ず V1.0.1 にバージョンアップしてから rm 削除コマンドを実行してください。V1.0.0 のバージョンでは、rm コマンドを実行する際に --Include および --exclude パラメータが機能せず、予期しない削除が発生する可能性があります。
- このコマンドでオブジェクトを削除するには、ポリシー許可 を設定する際に、actionを `cos:HeadBucket` 、 `cos:HeadObject` 、 `cos:GetBucket` 、 `cos:DeleteObject` 、 `cos:DeleteMultipleObjects` に設定する必要があります。
- このコマンドでバージョン履歴を削除するには（--all-versionsまたは--version-idを渡す場合）、ポリシー許可 を設定する際に、actionを `cos:HeadBucket` 、 `cos:HeadObject` 、 `cos:GetBucket` 、 `cos:DeleteObject` 、 `cos:DeleteMultipleObjects` 、 `cos:GetBucketVersioning` 、 `cos:GetBucketObjectVersions` に設定する必要があります。

詳細な権限については、CAM対応API をご参照ください。ご質問やご不明な点がございましたら、お問い合わせ ください。

## コマンド形式

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

rm コマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスするバケットを指定します。パラメータ設定 のバケットの別名、またはバケット名を使用してアクセスすることがサポートされております。バケット名でアクセスする場合は、追加の `endpoint flag` が必要になります。 | バケットの別名を使用してアクセスする場合：`cos://example-alias` バケット名を使用してアクセスする場合：`cos://examplebucket-1250000000` |

| | | |
|---|---|---|
| /prefix/ | オプションのパラメータ。特定のフォルダを指定します。 | /picture/ |

rmコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
|---|---|---|
| –h | ––help | ヘルプ情報を出力 |
| –c | ––config–path | 使用する設定ファイルパスを指定 |
| なし | ––include | 特定のモードを含むファイル |
| なし | ––exclude | 特定のモードを除外したファイル |
| –r | ––recursive | フォルダ内のすべてのファイルを再帰的にトラバーサル処理するかどうか |
| –f | ––force | 強制削除（ファイルを削除するまで確認情報はポップアップ表示されません） |
| なし | ––fail–output | このオプションは、ファイル削除時のエラー出力（デフォルトは `true` 、有効）を有効にするかどうかを決定します。有効にすると、ファイル転送の失敗が指定したディレクトリに記憶されます（指定しない場合はデフォルトで `./coscli_output` になる）。無効にすると、エラーファイルの数だけがコンソールに出力されます。 |
| なし | ––fail–output–path | このオプションは、ファイル削除時のエラー出力を記録するファイルの指定に使用されます。カスタムフォルダパスを指定することで、エラー出力フォルダの場所と名前を管理できます。このオプションが設定されていない場合は、デフォルトのエラーログフォルダ `./coscli_output` が使用されます。 |
| なし | ––all–versions | バージョン管理が有効になっているバケット内で、かつ––recursive（–r）パラメータを指定した場合にのみ使用可能で、指定されたパス下のすべてのバージョンを再帰的に削除します。 |
| なし | ––version–id | バージョン管理が有効になっているバケット内で、かつ––recursive（–r）パラメータを指定されてない場合にのみ使用可能で、指定されたオブジェクトの指定バージョンを削除します。 |

**説明：**

- `--include` と `--exclude` は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満たすファイルをフィルタリングすることができます。
- zshを使用する場合、pattern文字列の両端に二重引用符を付ける必要がある場合があります。

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

- このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

### fig1.pngファイルを削除します

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

### pictureフォルダからすべてのファイルを削除

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

### fig1.pngの指定バージョンを削除

```
./coscli rm cos://bucket1/fig1.png --version-id xxx
```

### testプレフィックスのすべてのバージョンを削除

```
./coscli rm cos://bucket1/test -r --all-versions
```

# ファイルハッシュ値の取得 – hash

最終更新日：：2025-05-19 15:52:08

hashコマンドは、ローカルファイルのハッシュ値を計算したり、Cloud Object Storage(COS)ファイルのハッシュ値を取得したりするために使います。

> **注意：**
> このコマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:HeadObject` に設定する必要があります。詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

```
./coscli hash <object-name> [flag]
```

hashコマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| <object-name> | アクセスするファイルを指定します。ローカルパスまたはCOSファイルパスを指定可能です。COSパスは、パラメータ設定 内のバケット別名またはバケット名からアクセスできます。バケット名からアクセスする場合は、 `endpoint` flagを追加する必要があります。 | ローカルパス：~/example.txt<br>バケット別名でCOSファイルパスを指定：cos://bucketalias/example.txt<br>バケット名でCOSファイルパスを指定：cos://examplebucket-1250000000/example.txt |

hashコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| -h | --help | ヘルプ情報を出力 |
| なし | --type | ハッシュタイプ（md5またはcrc64で、デフォルトはcrc64）<br>注意：MD5はetagのみを取得できます。ファイルのMD5を取得するには、ファイルをダウンロードしてから計算してください |

> **説明：**

> このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

### ローカルファイルのcrc64を計算します

```
./coscli hash ~/test.txt
```

### COSファイルのmd5を取得します

```
./coscli hash cos://bucket1/example.txt --type=md5
```

# マルチパートアップロード中に発生したフラグメントを一覧表示 – lsparts

最終更新日：：2025-05-19 15:52:09

lspartsコマンドは、マルチパートアップロード中に発生したフラグメントを一覧表示します。

> ⚠ 注意：
> - このコマンドを使用してチャンクアップロードタスクを表示するには、ポリシー許可を設定する際に、actionを `cos:PutBucket` に設定する必要があります。
> - このコマンドを使用してチャンクアップロードタスクに対応するファイルチャンクを表示するには、ポリシー許可を設定する際に、actionを `cos:ListMultipartUploads` 、 `cos:ListParts` に設定する必要があります。
>
> 詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

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

lspartsコマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスするバケットを指定します。パラメータ設定 のバケットの別名、またはバケット名を使用してアクセスすることがサポートされております。バケット名でアクセスする場合は、追加の `endpoint flag` が必要になります。 | バケットの別名を使用してアクセスする場合：`cos://example-alias` バケット名を使用してアクセスする場合：`cos://examplebucket-1250000000` |
| /prefix/ | オプションのパラメータ。特定のフォルダを指定します。 | /picture/ |

lspartsコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| –h | ––help | ヘルプ情報を出力 |

| | | |
|---|---|---|
| なし | --include | 特定のモードを含むファイル |
| なし | --exclude | 特定のモードを除外したファイル |
| なし | --limit | 一覧表示する最大数（0～1000）を指定 |
| なし | --upload-id | アップロード記録ID。uploadidを指定する場合、特定のファイルを指定する必要があり、uploadidに対応するファイルチャンク情報のみを表示します。uploadidが指定されていない場合、指定されたプレフィックスで進行中のチャンクアップロードタスクを表示します。 |

**説明：**

- チャンクアップロードの詳細については、チャンクアップロード をご参照ください。
- このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

### bucket1バケットのすべての進行中のチャンクアップロードタスクを表示

```
./coscli lsparts cos://bucket1
```

返された結果の例は以下の通りで、出力情報には次のものが含まれます：オブジェクトキー（ストレージバケット内のオブジェクトの唯一な識別子）、チャンクアップロードのID、チャンクアップロードの開始時間、バケット内のチャンク総数。

```
      KEY      |              UPLOAD ID              |      INITIATE
TIME
---------------+-------------------------------------+------------------
----------
    test.txt   | 1671191183635d2b71b1d68a0********** | 2022-01-
01T00:00:00.000Z
---------------+-------------------------------------+------------------
----------
                                                                TOTAL: 1
                                                    ------------------
----------
```

### bucket1バケットのtest.txtファイルのupload_idが 1671191183635d2b71b1d68a0**********のすべてのアップロードチャンクをリスト

```
./coscli lsparts cos://bucket1/test.txt --upload-
id="1671191183635d2b71b1d68a0**********"
```

返された結果の例は以下の通りで、出力情報には次のものが含まれます：PARTNUMBER（チャンク番号）、ETAG（ブロックのMD5）、LastModified（チャンクが最後に変更された時間）、SIZE（チャンクサイズ）。

```
  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


-------------
```

# フラグメントのクリーンアップ – abort

最終更新日：：2025-05-19 15:52:09

abortコマンドは、マルチパートアップロード中に発生したファイルフラグメントをクリーンアップするときに使います。

> ⚠ 注意：
> このコマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:ListMultipartUploads` 、 `cos:AbortMultipartUpload` に設定する必要があります。詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

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

abortコマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスするバケットを指定します。パラメータ設定 のバケットの別名、またはバケット名を使用してアクセスすることがサポートされております。バケット名でアクセスする場合は、追加の `endpoint flag` が必要になります。 | バケットの別名を使用してアクセスする場合：`cos://example-alias` バケット名を使用してアクセスする場合：`cos://examplebucket-1250000000` |
| /prefix/ | オプションのパラメータ。特定のフォルダを指定します。 | /picture/ |

abortコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| –h | ––help | ヘルプ情報を出力 |
| なし | ––include | 特定のモードを含むファイル |
| なし | ––exclude | 特定のモードを除外したファイル |
| なし | ––fail-output | このオプションは、ファイルチャンク削除時にエ |

| | | |
|---|---|---|
| | | ラー出力を有効にするかどうかを決定します（デフォルトは `true` 、有効）。有効にすると、ファイルチャンク削除が失敗した場合、指定されたディレクトリに記録されます（指定されていない場合、デフォルトは `./coscli_output` ）。無効にすると、失敗した数のみがコンソールに出力されます。 |
| なし | --fail-output-path | このオプションは、ファイルチャンク削除のエラー出力フォルダを指定するために使用されます。カスタムフォルダパスを提供することで、エラー出力フォルダの場所と名前を制御できます。このオプションが設定されていない場合、デフォルトのエラーログフォルダ `./coscli_output` が使用されます。 |

**説明：**
このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

### bucket1にあるすべてのファイルフラグメントをクリア

```
./coscli abort cos://bucket1
```

### bucket1のpictureフォルダにあるすべてのフラグメントをクリア

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

# アーカイブファイルの取得 – restore

最終更新日：：2025-05-19 15:52:09

restoreコマンドは、アーカイブファイルを取得するときに使います。

> **注意：**
> このコマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:HeadBucket`，`cos:GetBucket`，`cos:PostObjectRestore` に設定する必要があります。詳細な権限については、CAM対応APIをご参照ください。

## コマンド形式

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

restoreコマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスするバケットを指定します。パラメータ設定 のバケットの別名、またはバケット名を使用してアクセスすることがサポートされております。バケット名でアクセスする場合は、追加の `endpoint flag` が必要になります。 | バケットの別名を使用してアクセスする場合：`cos://example-alias` バケット名を使用してアクセスする場合：`cos://examplebucket-1250000000` |
| /prefix/ | オプションのパラメータ。特定のフォルダを指定します。 | /picture/ |

restoreコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| –h | ––help | ヘルプ情報を出力 |
| なし | ––include | 特定のモードを含むファイル |
| なし | ––exclude | 特定のモードを除外したファイル |
| –d | ––days | 一時ファイルの有効期限を指定（デフォルトは3日間） |

| | | |
|---|---|---|
| -m | --mode | リカバリモードを指定（デフォルトはStandard）<br>• アーカイブストレージタイプのデータを復元する場合、選択できる値はExpedited、Standard、Bulkであり、それぞれクイックリトリーブ、標準リトリーブ、およびバッチリトリーブに対応します。<br>• ディープアーカイブストレージタイプのデータを復元する場合、選択できる値はStandard、Bulkです |
| -r | --recursive | フォルダの再帰的なトラバーサル処理 |
| なし | --fail-output | このオプションは、ファイル復元失敗のエラー出力を有効にするかどうかを決定します（デフォルトは `true` 、有効）。有効にすると、ファイル回復失敗が指定されたディレクトリ内に記録されます（指定されていない場合、デフォルトは `./coscli_output` ）。無効にすると、エラーファイルの数のみがコンソールに出力されます。 |
| なし | --fail-output-path | このオプションは、ファイル復元失敗のエラー出力フォルダを指定するために使用されます。カスタムフォルダパスを提供することで、エラー出力フォルダの場所と名前を制御できます。このオプションが設定されていない場合、デフォルトのエラーログフォルダ `./coscli_output` が使用されます。 |

**説明:**

- `--include` と `--exclude` は標準的な正規表現の構文をサポートしており、これを使えば特定の条件を満たすファイルをフィルタリングすることができます。
- zshを使用する場合、pattern文字列の両端に二重引用符を付ける必要がある場合があります。

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

- このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。
- アーカイブファイルの取得の詳細は、POST Object restore をご参照ください。

## 操作事例

### 標準モードでbucket1のアーカイブファイルを取得します

```
./coscli restore cos://bucket1/picture.jpg
```

### bucket1内のpictureフォルダにあるすべてのアーカイブファイルを超高速モードで取得します

```
./coscli restore cos://bucket1/picture/ -r --mode Expedited
```

# ソフトリンクの作成/取得 – symlink

最終更新日：：2025-05-19 15:52:09

symlink コマンドは、オブジェクトへのソフトリンクを作成または取得するために使用されます。このソフトリンクを使用してオブジェクトをすばやく見つけることができます。

> ⚠ 注意：
> - ソフトリンク作成コマンドを使用するには、ポリシー許可 を設定する際に、actionを `cos:PutObject` に設定する必要があります。
> - ソフトリンク取得コマンドを使用するには、ポリシー許可 を設定する際に、actionを `cos:GetObject` に設定する必要があります。
>
> 詳細な権限については、CAM対応API をご参照ください。

## コマンドの形式

```
./coscli symlink --method create cos://<bucket-name>/<key> --link
linkKey [flag]
```

symlink コマンドには以下のパラメータが含まれています。

| パラメータの形式 | パラメータの用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | バケットを指定する必要があります。構成パラメータ 内のバケットエイリアスか、バケット名を使用してアクセスすることがサポートされています。バケット名を使用してアクセスする場合は、「endpoint」flag を追加する必要があります。 | バケットエイリアスを使用してアクセスする場合：cos://example-alias<br>バケット名を使用してアクセスする場合：cos://examplebucket-1250000000 |
| /<key> | 创ソフトリンクを作成するときにオブジェクト名を指定する必要があります。 | /test |

symlink コマンドには、次のオプションのflag が含まれています。

| flag の略称 | flag の全称 | flag の用途 |
| --- | --- | --- |
| –h | ––help | このコマンドの具体的な使い方を確認します。 |
| なし | ––method | オプションの値  create/get |
| | | |

| | | |
|---|---|---|
| なし | --link | ソフトリンクのkey 値 |

## 操作例

### bucket1 バケット内の picture.jpg のソフトリンク symlink を作成します。

```
./coscli symlink --method create cos://bucket1/picture.jpg --link
symlink
```

### ソフトリンク symlink で指定されるオブジェクトを取得します。

```
./coscli symlink --method get cos://bucket2 --link symlink
```

# オブジェクト内容の確認 – cat

最終更新日：：2025-05-19 15:52:09

cat コマンドは、オブジェクトの内容を表示するために使用されます。

> **注意：**
> このコマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:GetObject` に設定する必要があります。詳細な権限については、CAM対応APIをご参照ください。

## コマンドの形式

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

cat コマンドには以下のパラメータが含まれています。

| パラメータの形式 | パラメータの用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name>/<key> | バケットに格納するオブジェクトを指定します。 構成パラメータ 内のバケットエイリアスか、バケット名を使用してアクセスすることがサポートされています。 バケット名を使用してアクセスする場合は、endpoint flag を追加する必要があります。 | バケットエイリアスを使用してアクセスする場合：cos://example-alias/test.txt<br>バケット名を使用してアクセスする場合：cos://examplebucket-1250000000/test.txt |

cat コマンドには、以下のオプションのフラグが含まれています。

| flag の略称 | flag の全称 | flag の用途 |
| --- | --- | --- |
| –h | ––help | このコマンドの具体的な使い方を確認 |

## 操作例

### bucket1 バケット内の test.txt の内容を確認

```
./coscli cat cos://bucket1/test.txt
```

# 署名付きURLの取得 – signurl

最終更新日：：2025-05-19 15:52:10

signurlコマンドは、オブジェクトの事前署名付きURLを取得するために使い、このURLからオブジェクトに匿名でアクセスすることができます。

## コマンド形式

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

signurlコマンドには以下のパラメータが含まれています：

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name>/<key> | アクセスするバケットを指定します。パラメータ設定 のバケットの別名、またはバケット名を使用してアクセスすることがサポートされております。バケット名でアクセスする場合は、追加の `endpoint flag` が必要になります。 | バケットの別名を使用してアクセスする場合：`cos://example-alias/test.txt`<br>バケット名を使用してアクセスする場合：`cos://examplebucket-1250000000/test.txt` |

signurlコマンドには、以下のオプションflagが含まれます。

| flagの略称 | flagの正式名称 | flagの用途 |
| --- | --- | --- |
| –h | ––help | ヘルプ情報を出力 |
| –t | ––time | URLの有効期限を設定（デフォルトは1000s） |

> **説明：**
> このコマンドのその他の一般的なオプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、一般オプション ドキュメンテーションを参照してください。

## 操作事例

### bucket1内にあるpicture.jpgの事前署名付きURLを取得します

```
./coscli signurl cos://bucket1/picture.jpg
```

### bucket2内のpicture.jpgの事前署名付きURLを取得し、URLの有効期限を1314秒に設定します

```
./coscli signurl cos://bucket2/picture.jpg --time 1314
```

# ディレクトリ配下の内容のリストアップと統計 – lsdu

最終更新日：：2025-05-19 15:52:10

lsdu コマンドは、指定されたプレフィックスの現在のレベル配下の内容を取得し、各 object とディレクトリ内の object のサイズと数を統計するために使用されます。

> **注意：**
> このコマンドを使用するには、ポリシー許可 を設定する際に、actionを `cos:HeadBucket` 、`cos:GetBucket` に設定する必要があります。詳細な権限については、CAM対応API をご参照ください。

## コマンドの形式

```
./coscli lsdu cos://<bucket-name>[/prefix/] [flags]
```

lsdu コマンドには以下のパラメータが含まれています。

| パラメータの形式 | パラメータの用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスするバケットを指定します。構成パラメータ 内のバケットエイリアスか、バケット名を使用してアクセスすることがサポートされています。バケット名を使用してアクセスする場合は、「endpoint」flag を追加する必要があります。 | バケット名を使用してアクセスする場合：cos://example-alias<br>バケット名を使用してアクセスする場合：cos://examplebucket-1250000000 |
| /prefix/ | オプションのパラメータ。特定のフォルダを指定します。 | /picture/ |

lsdu コマンドには、以下のオプションの flag が含まれています。

| flag の略称 | flag の全称 | flag の用途 |
| --- | --- | --- |
| -h | --help | このコマンドの具体的な使い方を確認します。 |
| なし | --include | 特定のパターンのファイルを含みます。 |
| なし | --exclude | 特定のパターンのファイルを除外します。 |

## 操作例

### bucket1 バケットのルートディレクトリの統計情報を取得します。

```
./coscli lsdu cos://bucket1/
```

返される情報には、ディレクトリまたは object 名、オブジェクトの総数、合計サイズが含まれます。 以下に例を示します。

```
         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
```

### bucket1 バケットの picture ディレクトリの統計情報を取得します。

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

> **説明:**
> ファイル数が多いと、このコマンドは時間がかかりますので、バックグラウンドで実行することを推奨します。

# バージョン管理 – bucket-versioning

最終更新日：：2025-05-19 15:52:10

bucket-versioningコマンドはストレージバケットのバージョン管理機能を管理するために使用され、バージョン管理の状態を有効化、一時停止、確認することができます。

> ⚠ 注意：
> - バージョン管理の有効化/一時停止コマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:PutBucketVersioning` に設定する必要があります。
> - バケットのバージョン状態確認コマンドを使用するには、ポリシー許可を設定する際に、actionを `cos:GetBucketVersioning` に設定する必要があります。
>
> さらに多くの権限については、CAM対応APIを参照してください。

## 命令形式

以下のbucket-versioningコマンドは、バージョン管理の有効化と停止に使用されます：

```
./coscli bucket-versioning --method [method] cos://<bucket-name>
versioning
```

bucket-versioningおよびそのサブコマンドには以下の選択可能なflagが含まれています：

| flag 略語 | flag 全称 | flag 用途 |
| --- | --- | --- |
| -h | --help | コマンドの具体的な使用方法 |
| -c | --method | 選択可能な値put/get |

## 操作例

### ストレージバケットのバージョン管理を有効化

```
./coscli bucket-versioning --method put cos://examplebucket Enabled
```

### ストレージバケットのバージョン管理を一時停止

```
./coscli bucket-versioning --method put cos://examplebucket Suspended
```

## バケットのバージョン管理状態を表示

```
./coscli bucket-versioning --method get cos://examplebucket
```

# オブジェクトタグ – object-tagging

最終更新日：：2025-10-28 15:46:31

object-tagging コマンドは、オブジェクトタグの作成（変更）、照会、削除に使用されます。各オブジェクトは最大10個のタグを設定できます。

**注意：**

- オブジェクトタグを照会する場合、ポリシー許可 を設定する際に、actionを `cos:GetObjectTagging` に設定する必要があります。
- オブジェクトタグを設定するには、ポリシー許可 を設定する際に、actionを `cos:PutObjectTagging` に設定する必要があります。
- オブジェクトタグを削除する場合、ポリシー許可 を設定する際に、actionを `cos:DeleteObjectTagging` に設定する必要があります。

その他の権限付与については、CAM対応API をご参照ください。

## コマンド形式

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

object-taggingコマンドには以下のパラメータが含まれています。

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスしたいバケットを指定します。パラメータ設定 で設定したバケットエイリアス、またはバケット名でアクセスできます。バケット名でアクセスする場合は、追加で `endpoint` flagを指定する必要があります。 | バケットエイリアスでアクセス：<br>cos://example-alias<br>バケット名でアクセス：<br>cos://examplebucket-1250000000 |

object-taggingコマンドには、以下のオプションflagが含まれます。

| flag（短縮形） | flag（完全形） | flag（使用） |
| --- | --- | --- |
| -h | --help | コマンドの具体的な使用方法 |
| | | |

| | | |
|---|---|---|
| なし | --method | 実行する操作を指定します。put（タグ設定）、get（タグ照会）、delete（タグ削除）、add（タグ追加）が含まれます |
| なし | --version-id | 操作対象の特定バージョン |

**説明:**
このコマンドのその他の共通オプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、共通オプション ドキュメントをご参照ください。

## オブジェクトタグの変更

オブジェクトタグはドキュメント（Key-Value）で表されます。オブジェクトの所有者およびPutObjectTagging権限を持つユーザーのみが追加または変更でき、それ以外の場合はエラーコード403 AccessDeniedが返されます。

### コマンド形式

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

`key#value` はキーバリューペアを示し、キーとバリューは `#` で区切られます。オブジェクトにタグが設定されていない場合、このコマンドは指定されたタグを追加します。オブジェクトにすでにタグが設定されている場合、このコマンドは既存のタグを上書きします。

### 操作例

バケットエイリアスがexample-alias配下のobjectオブジェクトに2つのタグを設定します。1つ目のキーは1、バリューは111、2つ目のキーは2、バリューは222とします。コマンドは以下の通りです。

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

## オブジェクトタグの照会

### コマンド形式

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

### 操作例

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

以下の出力結果は、バケットエイリアスexample-alias配下のobjectオブジェクトに2つのタグが設定されていることを示します。1つ目のキーは1、バリューは111、2つ目のキーは2、バリューは222です。

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

## オブジェクトの指定タグの削除

### コマンド形式

```
./coscli object-tagging --method delete cos://bucketAlias/object
key1#value1
```

### 操作例

```
./coscli object-tagging --method delete cos://example-alias/object 1#111
```

## オブジェクトの全タグの削除

### コマンド形式

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

### 操作例

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

## オブジェクトタグの追加

オブジェクトタグはキーバリューペア（Key-Value）で表されます。オブジェクトの所有者およびPutObjectTagging権限を持つユーザーのみが追加または変更でき、それ以外の場合はエラーコード403 AccessDeniedが返されます。

## コマンド形式

```
./coscli object-tagging --method add cos://bucketAlias/object
key3#value3
```

`key#value` はキーバリューペアを示し、キーとバリューは `#` で区切られます。

## 操作例

バケットエイリアスがexample-alias配下のobjectオブジェクトに1つのタグを追加します。キーは1、バリューは111とします。コマンドは以下の通りです。

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

# バケットACLの管理 – bucket-acl

最終更新日：：2025-10-28 15:53:27

bucket-aclコマンドは、バケットのACLを設定、照会するために使用します。

> **注意：**
> - バケットのACLを照会する場合、ポリシー許可を設定する際に、actionを `cos:GetBucketACL` に設定する必要があります。
> - バケットのACLを設定する場合、ポリシー許可を設定する際に、actionを `cos:PutBucketACL` に設定する必要があります。
>
> その他の権限付与については、CAM対応APIをご参照ください。

## コマンド形式

```
./coscli bucket-acl --method [method] cos://<bucket-name>
```

bucket-aclコマンド、以下のパラメータが含まれます。

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスしたいバケットを指定します。パラメータ設定で設定したバケットエイリアス、またはバケット名でアクセスできます。バケット名でアクセスする場合は、追加で `endpoint` flagを指定する必要があります。 | バケットエイリアスでアクセス：cos://example-alias<br>バケット名でアクセス：cos://examplebucket-1250000000 |

bucket-aclコマンドには、以下のオプションflagが含まれます。

| flag（短縮形） | flag（完全形） | flag（使用） |
| --- | --- | --- |
| -h | --help | コマンドの具体的な使用方法 |
| なし | --method | 実行する操作を指定します。put（バケットACL設定）、get（バケットACL照会）が含まれます。 |
| なし | --acl | ファイルのACLを設定します。例えば、private、public-read、public- |

| | | |
|---|---|---|
| | | read-write。 |
| なし | --grant-read | 権限を付与されるユーザーに、バケットの読み取り権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |
| なし | --grant-read-acp | 権限を付与されるユーザーに、バケットのアクセス制御リスト（ACL）の読み取り権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |
| なし | --grant-write-acp | 権限を付与されるユーザーに、バケットのアクセス制御リスト（ACL）の書き込み権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |
| なし | --grant-full-control | 権限を付与されるユーザーに、バケットを操作するすべての権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |

**説明：**
このコマンドのその他の共通オプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、共通オプション ドキュメントをご参照ください。

## バケットACLの設定

### 操作例

ユーザー100000000013および100000000012に、バケットエイリアスがexample-aliasのバケットに対する読み取り権限を付与します。コマンドは以下の通りです。

```
./coscli bucket-acl --method put cos://example-alias --grant-
read="id=\"100000000013\",id=\"100000000012\""
```

## バケットACLの照会

### 操作例

バケットエイリアスがexample-aliasの権限リストを照会します。

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

以下の結果が出力されます。

```
  SECTION   | KEY           | VALUE
------------+---------------+-------------------------------------------------
-
  Owner     | UIN           |
+           +---------------+-------------------------------------------------
+
            | ID            | qcs::cam::uin/100000000:uin/100000000
+           +---------------+
+
            | Display Name  |
+-----------+---------------+-------------------------------------------------
+
            |               |
+-----------+---------------+-------------------------------------------------
+
  Grant #1  | Permission    | READ
+           +---------------+-------------------------------------------------
+
            | Grantee Type  | CanonicalUser
+           +---------------+-------------------------------------------------
+
            | ID            | qcs::cam::uin/100000000013:uin/100000000013
+           +---------------+
+
            | 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/100000000:uin/100000000 (UIN: )
 - Total Grants: 2
 - Permissions:
   - READ: 2 grants
```

# オブジェクトACLの管理 – object-acl

最終更新日：：2025-10-28 15:56:22

object-aclコマンドは、オブジェクトのACLを設定、照会するために使用します。

**注意：**
- オブジェクトのACLを照会する場合、ポリシー許可を設定する際に、actionを `cos:GetObjectACL` に設定する必要があります。
- オブジェクトのACLを設定する場合、ポリシー許可を設定する際に、actionを `cos:PutObjectACL` に設定する必要があります。

その他の権限付与については、CAM対応APIをご参照ください。

## コマンド形式

```
./coscli object-acl --method [method] cos://<bucket-name>/object
```

object-aclコマンドには、以下のパラメータが含まれます。

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスしたいバケットを指定します。パラメータ設定で設定したバケットエイリアス、またはバケット名でアクセスできます。バケット名でアクセスする場合は、追加で `endpoint` flagを指定する必要があります。 | バケットエイリアスでアクセス：cos://example-alias<br>バケット名でアクセス：cos://examplebucket-1250000000 |

object-aclコマンドには、以下のオプションflagが含まれます。

| flag（短縮形） | flag（完全形） | flag（使用） |
| --- | --- | --- |
| -h | --help | コマンドの具体的な使用方法 |
| なし | --method | 実行する操作を指定します。put（オブジェクトACL設定）、get（オブジェクトACL照会）が含まれます |
| なし | --acl | ファイルのACLを設定します。例：private、public-read |

| | | |
|---|---|---|
| なし | --grant-read | 権限を付与されるユーザーに、オブジェクトの読み取り権限を与えます。形式は id="[OwnerUin]" です。例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |
| なし | --grant-read-acp | 権限を付与されるユーザーに、オブジェクトのアクセス制御リスト（ACL）の読み取り権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |
| なし | --grant-write-acp | 権限を付与されるユーザーに、オブジェクトのアクセス制御リスト（ACL）の書き込み権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |
| なし | --grant-full-control | 権限を付与されるユーザーに、オブジェクトを操作するすべての権限を与えます。形式はid="[OwnerUin]"、例：id="100000000001"。半角カンマ（,）で区切って複数のユーザーを指定できます。例：id="100000000001",id="100000000002"。 |

**説明：**
このコマンドのその他の共通オプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、共通オプションドキュメントをご参照ください。

## 操作例

### オブジェクトACLの設定

ユーザー100000000013および100000000012に、バケットエイリアスexample-alias配下のobjectオブジェクトに対する読み取り権限を付与します。コマンドは以下の通りです。

```
./coscli object-acl --method put cos://example-alias/object --grant-read="id=\"100000000013\",id=\"100000000012\""
```

### オブジェクトACLの照会

バケットエイリアスexample-alias配下のobjectオブジェクトの権限リストを照会します。

```
./coscli object-acl --method get cos://example-alias/object
```

以下の結果が出力されます。

```
 SECTION   | KEY           | VALUE
-----------+---------------+-------------------------------------------------
-
  Owner    | UIN           |
+          +---------------+-------------------------------------------------
+
           | ID            | qcs::cam::uin/1000000000:uin/1000000000
+          +---------------+
+
           | Display Name  |
+----------+---------------+-------------------------------------------------
+
           |               |
+----------+---------------+-------------------------------------------------
+
  Grant #1 | Permission    | READ
+          +---------------+-------------------------------------------------
+
           | Grantee Type  | CanonicalUser
+          +---------------+-------------------------------------------------
+
           | ID            | qcs::cam::uin/100000000013:uin/100000000013
+          +---------------+
+
           | 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
```

# バケットポリシー – bucket-policy

最終更新日：：2025-10-28 16:00:59

bucket-policyコマンドは、バケットポリシーの設定、照会、削除に使用します。

**注意：**

- バケットポリシーを照会する場合、ポリシー許可を設定する際に、actionを `cos:GetBucketPolicy` に設定する必要があります。
- バケットポリシーを設定する場合、ポリシー許可を設定する際に、actionを `cos:PutBucketPolicy` に設定する必要があります。
- バケットポリシーを削除する場合、ポリシー許可を設定する際に、actionを `cos:DeleteBucketPolicy` に設定する必要があります。

その他の権限付与については、CAM対応APIをご参照ください。

## コマンド形式

```
./coscli bucket-policy --method [method] cos://<bucket-name>
```

bucket-policyコマンドには、以下のパラメータが含まれます。

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスしたいバケットを指定します。パラメータ設定で設定したバケットエイリアス、またはバケット名でアクセスできます。バケット名でアクセスする場合は、追加で `endpoint` flagを指定する必要があります。 | バケットエイリアスでアクセス：cos://example-alias<br>バケット名でアクセス：cos://examplebucket-1250000000 |

bucket-policyコマンドには、以下のオプションflagが含まれます。

| flag（短縮形） | flag（完全形） | flag（使用） |
| --- | --- | --- |
| -h | --help | コマンドの具体的な使用方法 |
| なし | --method | 実行する操作を指定します。put（バケットポリシー設定）、get（バケットポリシー照会）、delete（バケットポリシー削除）が含まれます |
| なし | --policy | バケットポリシー（JSON形式または直接ファイルパスを指定します。 |

| | | |
|---|---|---|
| | | 例：/data/policy.txtは `file:///data/policy.txt` と指定する必要があります) |

**説明:**
このコマンドのその他の共通オプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、共通オプションドキュメントをご参照ください。

## バケットポリシーの設定

### 操作例

バケットエイリアスがexample-aliasのバケットポリシーを設定します。コマンドは以下の通りです（バケットポリシーの設定をご参照ください）。

```
./coscli bucket-policy --method put cos://example-alias --policy="
{\"Statement\":[...]}"
```

## バケットポリシーの照会

### 操作例

バケットエイリアスがexample-aliasのポリシー情報を照会します。

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

以下の結果が出力されます。

```
  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
```

## バケットポリシーの削除

### 操作例

バケットエイリアスがexample-aliasのバケットポリシーを削除します。コマンドは以下の通りです。

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

# バケット暗号化ポリシー – bucket–encryption

最終更新日：：2025–10–28 16:03:25

bucket–encryption コマンドは、バケットの暗号化ポリシーを設定、照会、削除するために使用します。

**注意：**

- バケットの暗号化ポリシーを照会する場合、ポリシー許可 を設定する際に、actionを `cos:GetBucketEncryption` に設定する必要があります。
- バケットの暗号化ポリシーを設定する場合、ポリシー許可 を設定する際に、actionを `cos:PutBucketEncryption` に設定する必要があります。
- バケットの暗号化ポリシーを削除する場合、ポリシー許可 を設定する際に、actionを `cos:DeleteBucketEncryption` に設定する必要があります。

その他の権限付与については、CAM対応API をご参照ください。

## コマンド形式

```
./coscli bucket-encryption --method [method] cos://<bucket-name>
```

bucket–encryptionコマンドには、以下のパラメータが含まれます。

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket–name> | アクセスしたいバケットを指定します。パラメータ設定 で設定したバケットエイリアス、またはバケット名でアクセスできます。バケット名でアクセスする場合は、追加で `endpoint` flagを指定する必要があります。 | バケットエイリアスでアクセス：cos://example–alias<br>バケット名でアクセス：cos://examplebucket–1250000000 |

bucket–encryptionコマンドには、以下のオプションflagが含まれます。

| flag（短縮形） | flag（完全形） | flag（使用） |
| --- | --- | --- |
| –h | ––help | コマンドの具体的な使用方法 |
| なし | ––method | 実行する操作を指定します。put（バケット暗号化ポリシー設 |

| | | |
|---|---|---|
| | | 定）、get（バケット暗号化ポリシー照会）、delete（バケット暗号化ポリシー削除）が含まれます |
| なし | --sse-algorithm | 暗号化アルゴリズム（AES256、SM4、KMS） |
| なし | --kms-master-key-id | KMS マスターキー ID |
| なし | --kms-algorithm | KMS 暗号化アルゴリズム（AES256、SM4） |

**説明：**
このコマンドのその他の共通オプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、共通オプション ドキュメントをご参照ください。

## バケット暗号化ポリシーの設定

### 操作例

バケットエイリアスがexample-aliasのバケット暗号化ポリシーを設定します。コマンドは以下の通りです。

```
./coscli bucket-encryption --method put cos://example-alias --sse-algorithm KMS
```

## バケット暗号化ポリシーの照会

### 操作例

バケットエイリアスがexample-aliasの暗号化ポリシー情報を照会します。

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

以下の結果が出力されます。

```
   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)
```

## バケット暗号化ポリシーの削除

### 操作例

バケットエイリアスがexample-aliasのバケット暗号化ポリシーを削除します。コマンドは以下の通りです。

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

# インベントリ – inventory

最終更新日：：2025-10-28 16:07:42

inventory コマンドは、インベントリの設定、照会、削除、一覧表示、および1回限りのインベントリの開始に使用します。

**注意：**

- インベントリを照会する場合、ポリシー許可 を設定する際に、actionを `cos:GetBucketInventory` に設定する必要があります。
- インベントリを設定する場合、ポリシー許可 を設定する際に、actionを `cos:PutBucketInventory` に設定する必要があります。
- インベントリを削除する場合、ポリシー許可 を設定する際に、actionを `cos:DeleteBucketInventory` に設定する必要があります。
- インベントリを一覧表示する場合、ポリシー許可 を設定する際に、actionを `cos:GetBucketInventory` に設定する必要があります。
- 1回限りのインベントリを開始する場合、ポリシー許可 を設定する際に、actionを `cos:PostBucketInventory` に設定する必要があります。

その他の権限付与については、CAM対応API をご参照ください。

## コマンド形式

```
./coscli inventory --method [method] cos://<bucket-name>
```

inventoryコマンドには、以下のパラメータが含まれます。

| パラメータ形式 | パラメータ用途 | 例 |
| --- | --- | --- |
| cos://<bucket-name> | アクセスしたいバケットを指定します。パラメータ設定 で設定したバケットエイリアス、またはバケット名でアクセスできます。バケット名でアクセスする場合は、追加で `endpoint` flagを指定する必要があります。 | バケットエイリアスでアクセス：<br>cos://example-alias<br>バケット名でアクセス：<br>cos://examplebucket-1250000000 |

inventory コマンドには、以下のオプションflagが含まれます。

| flag（短縮形） | flag（完全形） | flag（使用） |
| --- | --- | --- |

| | | |
|---|---|---|
| -h | --help | コマンドの具体的な使用方法 |
| なし | --method | 実行する操作を指定します。put（バケットインベントリ設定）、get（バケットインベントリ照会）、delete（バケットインベントリ削除）、list（バケットインベントリ一覧表示）、post（1回限りのインベントリ開始）が含まれます。 |
| なし | --task-id | インベントリタスクID |
| なし | --configuration | インベントリ設定（JSON形式、XML形式、または直接ファイルパスを指定できます。例：/data/configuration.txt は `file:///data/configuration.txt` と指定する必要があります） |

**説明：**
このコマンドのその他の共通オプション（バケットの切り替え、ユーザーアカウントの切り替えなど）については、共通オプション ドキュメントをご参照ください。

## バケットインベントリの設定

### 操作例

バケットエイリアスがexample-aliasのインベントリを設定します。コマンドは以下の通りです（具体的な形式については、バケットインベントリの設定 をご参照ください）。

```
./coscli inventory --method put cos://example-alias --task-id list4 --
configuration "<InventoryConfiguration>...</InventoryConfiguration>"
```

## バケットインベントリの照会

### 操作例

バケットエイリアスがexample-aliasのlist4インベントリタスクを照会します。

```
./coscli inventory --method get cos://example-alias --task-id list4
```

以下の結果が出力されます。

```
  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
```

## バケットインベントリの削除

### 操作例

バケットエイリアスがexample-aliasのバケットのlist4インベントリタスクを削除します。コマンドは以下の通りです。

```
./coscli inventory --method delete cos://example-alias --task-id list4
```

## すべてのバケットインベントリの一覧表示

### 操作例

バケットエイリアスがexample-aliasのすべてのインベントリタスクを一覧表示します。

```
./coscli inventory --method list cos://example-alias
```

以下の結果が出力されます。

```
  ID     | STATUS    | SCHEDULE           | INCLUDEDOBJECTVERSIONS |
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: 1000000000                                |
| 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: 1000000000                                  | Period: 2026-01-
18 to 2019-09-17 | StorageClass
        |          |                    |                          |
Prefix: list4                                        |
| IsMultipartUploaded
        |          |                    |                          |
|                                    | Tag
        |          |                    |                          |
|                                    | LastModifiedDate
--------+----------+--------------------+--------------------------+-------
-------------------------------------------------+----------------------
-----------+----------------------
Detailed COS Bucket Inventory Configurations

Total inventory configurations: 3
```

## 1回限りのインベントリを開始

### 操作例

バケットエイリアスexample-aliasに対し、1回限りのインベントリタスクを開始します。コマンドは以下の通りです（具体的な形式については、1回限りのインベントリタスクの開始 をご参照ください）。

```
./coscli inventory --method post cos://example-alias --task-id list4 --
configuration "<InventoryConfiguration>...</InventoryConfiguration>"
```

# よくある質問

最終更新日：：2024-06-26 10:27:35

## COSCLIツールとCOSCMDツールの違いとは何ですか。

1. COSCLIツールはgolangでビルドされており、コンパイルされたバイナリーパッケージを直接配布します。ユーザーはインストールやデプロイの際に依存関係をインストールする必要がなく、開梱したらすぐに使用可能です。COSCMDツールはPythonでビルドされており、ユーザーはインストールする前に、Python環境と依存パッケージをインストールする必要があります。
2. COSCLIツールはバケットエイリアスの設定に対応しており、`<BucketName-APPID>` の代わりに短い文字列を使用できるため、ユーザーにとって使い勝手がよいです。COSCMDツールはバケットエイリアスの設定に対応していないため、ユーザーは `<BucketName-APPID>` と入力してバケットを指定し、複雑で読みにくいコマンドに対応しなければなりません。
3. COSCLIツールは設定ファイル内に複数のバケットの設定やバケット間操作にも対応しています。COSCMDツールは設定ファイルに1つのバケットしか設定できず、バケット間の操作コマンドは冗長になりすぎます。

# COSCMDツール

最終更新日：：2025-07-21 16:18:01

## 機能説明

ユーザーはCOSCMDツールを使用すれば、簡単なコマンドラインによって、オブジェクト(Object)の一括アップロード・ダウンロード・削除などの操作が行えます。

## 使用環境

### システム環境

Windows、Linux 、macOSシステムをサポートします。

> **説明：**
> - ローカル文字形式がUTF-8であることを確認してください。UTF-8でない場合、中国語版のファイルを操作すると、異常が発生します。
> - 本機の時刻が協定世界時で修正されていることを確認してください。誤差が大きすぎると、正常に使用できなくなります。

### ソフトウェア依存

- Python 2.7/3.5/3.6/3.9。
- 最新バージョンのpip。

> **説明：**
> pipが統合された最新版のPython（例：バージョン3.9.0）を直接インストールすることをお勧めします。

### インストールおよび設定

- 環境のインストールと設定操作の詳細については、Pythonのインストールと設定 をご参照ください。
- pip環境のインストールと設定操作の詳細については、公式サイトpipのインストールの説明 をご参照ください。

## ダウンロードとインストール

ユーザーがCOSCMDをインストールする方法として、以下の3つの方法が提供されています。

### 1.1 pipによるインストール

`pip` コマンドを実行してインストールします。

```
pip install coscmd
```

インストールの成功後、ユーザーは `-v` または `--version` コマンドを使用して、現在のバージョン情報を確認することができます。

> ⚠ 注意：
> Windowsでインストールした場合は、環境変数に `C:\python_install_dir;` と `C:\python_install_dir\Scripts` という2つのパスを追加する必要があります。

### 1.2 pipの更新

インストールが完了したら、次のコマンドを実行して更新します。

```
pip install coscmd -U
```

## 2. ソースコードのインストール（非推奨）

ソースコードダウンロードアドレス：ここをクリック 。

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

> ⚠ 注意：
> Pythonバージョンが2.6で、pipの依存ライブラリへのインストールが失敗しやすい場合は、この方法でインストールすることをお勧めします。

## 3. オフラインインストール

> ⚠ 注意：
> 2台のマシンのPythonのバージョンが同じであることを確認してください。同じでない場合、インストールは失敗してしまいます。

```
# パブリックネットワークを備えたマシンで次のコマンドを実行します
mkdir coscmd-packages
pip download coscmd -d coscmd-packages
tar -czvf coscmd-packages.tar.gz coscmd-packages
```

```
# インストールパッケージをパブリックネットワークのないマシンにコピーしてから、次のコマンドを実行します
tar -xzvf coscmd-packages.tar.gz
pip install coscmd --no-index -f coscmd-packages
```

## パラメータの設定

### helpオプションの確認

ユーザーは、`-h` または `--help` コマンドを使用して、ツールのhelp情報や使用方法を確認することができます。

```
coscmd -h
```

help情報は次のとおりです。

```
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,restore,signurl,createbucket,deletebucket,putobjectacl,getobjectacl,putbucketacl,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,restore,signurl,createbucket,deletebucket,putobjectacl,getobjectacl,putbucketacl,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
```

これに加えて、ユーザーは各コマンドの後に（パラメータなしで） `-h` を入力すると、そのコマンドの具体的な使用法を確認することができます。次に例を示します。

```
coscmd upload -h  //コマンドの使用方法を確認します
```

## configコマンドを使用して設定ファイルを発行

> **注意：**
> - ユーザーには一時キーを使用してSDKを呼び出し、一時権限承認方式によってSDK使用の安全性をさらに向上させることをお勧めします。一時キーを申請する際は、最小権限の原則についてのガイドに従い、ターゲットバケットまたはオブジェクト以外のリソースが漏洩しないようにしてください。
> - どうしてもパーマネントキーを使用したい場合は、最小権限の原則についてのガイドに従って、パーマネントキーの権限範囲を限定することをお勧めします。

configコマンドは、 `~/.cos.conf` に設定ファイルを自動的に発行します。コマンド形式は次のとおりです。

```
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]
```

> **説明：**
> ここで「[]」内のフィールドはオプション、「<>」内のフィールドは入力が必要なパラメータです。

パラメータ設定の説明は次のとおりです。

| オプション | パラメータ説明 | 有効値 | 入力必須かどうか |
| --- | --- | --- | --- |
| -a | キーIDはAPIキーコンソールに移動して取得してください | 文字 | はい |

| | | | |
|---|---|---|---|
| | | 列 | |
| –s | KeyはAPIキーコンソールに移動して取得します | 文字列 | はい |
| –t | 一時キーtokenは、一時キーを使用するときに設定が必要で、x–cos–security–tokenヘッダーを設定します | 文字列 | いいえ |
| –b | 指定されたバケット名。バケットの命名形式はBucketName–APPIDです。命名ルールをご参照ください。初回設定時に使用する場合、COSコンソールでバケットを作成し、設定ツールとして用いる必要があります | 文字列 | はい |
| –r | バケットの所在リージョンです。リージョンとアクセスドメイン名をご参照ください。 | 文字列 | はい |
| –e | リクエストのENDPOINTを設定します。ENDPOINTパラメータを設定すると、REGIONパラメータは無効になります。デフォルトのドメイン名を使用している場合、ここでの設定形式は、`cos.<region>.myqcloud.com`となります。グローバルアクセラレーションドメイン名を使用する場合、設定は`cos.accelerate.myqcloud.com`となります | 文字列 | いいえ |
| –m | マルチスレッド操作の最大スレッド数（デフォルトは5、範囲は1～30） | 数値 | いいえ |
| –p | チャンク操作の1チャンクサイズ（MB単位、デフォルトは1MB、範囲は1～1000） | 数値 | いいえ |
| ––do–not–use–ssl | HTTPSではなく、HTTPプロトコルを使用します | 文字列 | いいえ |
| ––anonymous | 匿名操作（署名なし） | 文字列 | いいえ |

configコマンドの使用例は次のとおりです。

```
coscmd config -a AChT4ThiXAbpBDEFGhT4ThiXAbp**** -s
WE54wreefvds3462refgwewe**** -b configure-bucket-1250000000 -r ap-
chengdu
```

## 設定ファイルの発行

COSCMDツールは、実行前にまず実行時に必要な情報を設定ファイルから読み込みます。COSCMDは、デフォルトでは `~/.cos.conf` から設定項目を読み込みます。

> **説明:**
> 設定する前に、COSコンソールでパラメータ設定用のバケットを作成し（例：configure-bucket-1250000000）、キー情報を作成する必要があります。

設定ファイルの例を次に示します。

```
[common]
secret_id = AKIDA6wUmImTMzvXZNbGLCgtusZ2E8mG****
secret_key = TghWBCyf5LIyTcXCoBdw1oRpytWk****
bucket = configure-bucket-1250000000
region = ap-chengdu
max_thread = 5
part_size = 1
retry = 5
timeout = 60
schema = https
verify = md5
anonymous = False
```

> **説明:**
> - 設定ファイルの `schema` 項目で、オプション値はhttp、https、デフォルトはhttpsです。
> - 設定ファイルの `anonymous` 項目で、オプション値はTrue、False、匿名モードを使用するかどうか、つまり署名を空にするかどうかを示します。
> - パラメータ設定の詳細については、コマンド `coscmd config -h` を使用して確認してください。

# 一般的なコマンド

## BucketとRegionのコマンドを指定

ユーザーがコマンドを実行するバケット名と所属リージョンを指定しない場合、パラメータの設定時に入力されたバケットがデフォルトで有効になります。異なるバケットで操作を実行する必要がある場合は、バケット名と所属リージョンを指定する必要があります。

> **説明:**
> - `-b <BucketName-APPID>` パラメータでバケット名を指定します。バケットの命名形式は

BucketName-APPIDです。ここに入力するバケット名は、必ずこの形式である必要があります。
- `-r <region>` でRegionを指定すると、バケットの所属リージョンを指定することができます。

- コマンド形式

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

- 操作事例 – バケット名がexamplebucket、所属リージョンが北京のバケットを作成します

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

- 操作事例 – Dドライブのファイルpicture.jpgを examplebucketという名前のバケットにアップロードします

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

## 設定ファイルとログファイルパスを指定するコマンド

ユーザーが設定ファイルのパスを指定しない場合、デフォルトの設定ファイルパス `~/.cos.conf` が使用されます。ログファイルパスが指定されない場合、デフォルトのログファイルパス `~/.cos.log` が使用されます。

**説明:**
- `-c <conf_path>` パラメータで設定ファイルパスを指定すると、COSCMDは実行時にこのパスから設定情報を読み込むようになります。
- `-l <log_conf>` パラメータでログパスを指定すると、COSCMDは実行中に生成されたログをこのパスのログファイルに出力します。

- コマンド形式

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

- 操作事例 – 設定ファイルパスを/data/home/cos_conf 、ログ出力パスを/data/home/cos_logと指定し、バケット名がexamplebucket、所属リージョンが北京のバケットを作成します

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

### Debugモード実行コマンド

各コマンドの前に `-d` または `--debug` を追加すると、コマンドの実行中に詳細な操作情報が表示されます。次に例を示します。

- コマンド形式

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

- 操作事例 – アップロード時に詳細情報を出力します

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

### Silenceモード実行コマンド

各コマンドの前に `-s` または `--silence` を追加すると、コマンドの実行中にいかなる情報も出力されなくなります。

> **説明:**
> このコマンドは、最小バージョン1.8.6.24を満たす必要があります。

- コマンド形式

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

- 操作事例

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

## 一般的なバケットコマンド

### バケットの作成

> **説明:**
> バケットを作成するコマンドを実行するときは、バケット名を指定するパラメータ `-b <BucketName-APPID>` と所属リージョンを指定するパラメータ `-r <Region>` `を付けてください。coscmd createbucketを直接実行した場合、バケット名と所属リージョンを指定しないと、既存のバケット（パラメータ設定時に入力したバケット）を作成したのと同じことになるため、エラーが発生します。

- コマンド形式

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

- 操作事例–バケット名がexamplebucket、所属リージョンが北京のバケットを作成します

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

### バケットの削除

> **説明:**
> `coscmd deletebucket` の使用法は、パラメータを設定する際にストレージバケットに対してのみ有効です。 `-b <BucketName-APPID>` でBucketを指定し、 `-r <region>` でRegionを指定することをお勧めします。

- コマンド形式

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

- 操作事例 – 空のバケットを削除します

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

- 操作事例 – 空ではないバケットを強制的に削除します

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

> **注意:**
> `-f` パラメータを使用すると、すべてのファイルやバージョン管理を有効にした後の履歴フォルダ、アップロードによって生成されたフラグメントを含むバケットが強制的に削除されますので、操作は慎重に行ってください。

## 一般的なオブジェクトコマンド

### ファイルのアップロード

- ファイルアップロードのコマンド形式

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

> ⚠ 注意：
> 「<>」のパラメータ を、アップロードする必要のあるローカルファイルパス(localpath)とCOSのストレージパス(cospath)に置き換えてください。

- 操作事例 – Dドライブのpicture.jpgファイルをCOSのdocディレクトリにアップロードします

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

- 操作事例 – Dドライブのdocフォルダからpicture.jpgファイルをCOSのdocディレクトリにアップロードします

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

- 操作事例 – オブジェクトタイプを指定し、アーカイブタイプのファイルをCOSのdocディレクトリにアップロードします

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

> ⚠ 注意：
> –Hパラメータを使用してHTTP headerを設定する場合は、形式がJSONであることを必ず確認してください。例：
> `coscmd upload -H "{'x-cos-storage-class':'Archive','Content-Language':'zh-CN'}" <localpath> <cospath>`
> 。その他のヘッダーについては、PUT Objectのドキュメントをご参照ください。

- 操作事例 – metaメタ属性を設定し、COSのdocディレクトリにファイルをアップロードします

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

## フォルダのアップロード

- フォルダアップロードのコマンド形式

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

> ⚠ 注意：
> Windowsユーザーは、システムに標準搭載されているcmdツールやPowerShellでCOSCMDのuploadコマンドを使用することをお勧めします。その他のツール（git bashなど）は、コマンドパスの解析ポリシーがPowerShellと異なるため、ユーザーのファイルが誤ったパスにアップロードされる可能性があります。

- 操作事例 – Dドライブのdocフォルダとそのファイルを COSのルートパスにアップロードします

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

- 操作事例 – Dドライブのdocフォルダとそのファイルを COSのdocパスにアップロードします

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

- 操作事例 – 同時アップロード、md5と同名・同サイズのファイルをスキップします

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

> ⚠ 注意：
> –sパラメータを使用すれば、同時アップロードによってmd5と一致するファイルのアップロードをスキップできます（COSの元のファイルは、必ずバージョン1.8.3.2以降のCOSCMDによってアップロードされている必要があります。デフォルトではx-cos-meta-md5のheaderが使用されます）。

- 操作事例 – 同時アップロード、同名・同サイズのファイルをスキップします

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

> ⚠ 注意：
> –sパラメータを使用すれば、同時アップロードを使用することができます。また、––skipmd5パラメータを使用すると、同名ファイルのサイズのみが比較され、同じサイズであればアップロードがスキップされます。

- 操作事例 – 同時アップロードを行い、「Dドライブのdocフォルダにある削除されたファイル」を削除します

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

- 操作事例 – Dドライブのdocフォルダにある.txtおよび.doc拡張子を持つファイルのアップロードを無視することを選択します

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

- 操作事例 – Dドライブのdocフォルダにある.txt拡張子を持つファイルのアップロードを無視することを選択します

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

> ⚠ 注意:
> - フォルダをアップロードするときに、`--ignore` パラメータを使用すると、いずれかのタイプのファイルを無視することができます。`--include` パラメータを使用すると、いずれかのタイプのファイルをフィルタリングすることができます。shellワイルドカードルールや複数のルールをサポートし、カンマ `,` で区切ります。特定のサフィックスを無視する場合は、必ず `""` で囲む必要があります。
> - `--ignore` を使用して特定のフォルダ内のすべてのファイルをフィルタリングしたい場合は、絶対パスを使用し、パスの前後に `""` を追加する必要があります。例えば、`coscmd upload -rs D:/doc / --ignore "D:/doc/ignore_folder/*"` などとします。

- 操作事例 – Dドライブのdocフォルダにある.txtと.docという拡張子を持つファイルをアップロードします

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

- 操作事例 – Dドライブのdocフォルダにある.txt拡張子を持つファイルをアップロードします

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

> ⚠ 注意:
> - 10MB以上のファイルをアップロードする場合、COSCMDはマルチパートアップロード方式を採用します。コマンドの使用法は単純なアップロードと同じで、`coscmd upload <localpath> <cospath>` です。
> - COSCMDは、大きなファイルのブレークポイントアップロード機能をサポートしています。大きな

ファイルのマルチパートアップロードが失敗した場合、このファイルの再アップロードでは失敗したチャンクのみがアップロードされ、最初からやり直すことはありません（再アップロードしたファイルのディレクトリとコンテンツがアップロードしたディレクトリと同じであることを確認してください）。
- COSCMDのマルチパートアップロードのときは、チャンクごとにMD5チェックが行われます。
- COSCMDのアップロードはデフォルトで `x-cos-meta-md5` というヘッダーが付きます。これはこのファイルのmd5値となりますが、--skipmd5パラメータがある場合、 このヘッダーは付きません。

## ファイルリストの照会

照会コマンドは次のとおりです。

- コマンド形式

```
coscmd list <cospath>
```

- 操作事例 – このバケット内のdoc/というプレフィックスを持つすべてのファイルリストを再帰的に照会します

```
coscmd list doc/
```

- 操作事例 – このバケット内のすべてのファイルリスト、ファイル数およびファイルサイズを再帰的に照会します

```
coscmd list -ar
```

- 操作事例 – examplefolderというプレフィックスを持つすべてのファイルリストを再帰的に照会します

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

- 操作事例 – このバケット内のすべてのファイル履歴を再帰的に照会します

```
coscmd list -v
```

**説明：**
- 「<>」内のパラメータを、ファイルリストを照会する必要のあるCOS上のファイルのパス(cospath)に置き換えてください。 `<cospath>` が空の場合は、デフォルトで現在のバケットルートディレクトリを照会します。

- `-a` を使用してすべてのファイルを照会します。
- `-r` を使用して再帰的に照会すると、リストアップされたファイルの数とサイズの合計が末尾に返されます。
- `-n num` を使用して、照会の最大値を設定します。

## ファイル情報の確認

コマンドは次のとおりです。

- コマンド形式

```
coscmd info <cospath>
```

- 操作事例 – doc/picture.jpgのメタ情報を確認します

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

**説明:**
「<>」内のパラメータを、表示させる必要のあるCOS上のファイルのパス(cospath)に置き換えてください。

## ファイルまたはフォルダのダウンロード

### ファイルダウンロードのコマンド形式

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

**注意:**
「<>」のパラメータ を、ダウンロードする必要のあるCOS上のファイルのパス(cospath)とローカルストレージパス(localpath)に置き換えてください。

- 操作事例 – COS上のdoc/picture.jpgをD:/picture.jpgにダウンロードします

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

- 操作事例 – COS上のdoc/picture.jpgをDドライブにダウンロードします

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

- 操作事例 – バージョンID付きのpicture.jpgファイルをDドライブにダウンロードします

```
coscmd download picture.jpg --versionId MTg0NDUxMzc2OTM4NTExNTg7Tjg
D:/
```

### フォルダダウンロードのコマンド形式

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

- 操作事例 – docディレクトリをD:/folder/docにダウンロードします

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

- 操作事例 – ルートディレクトリファイルをダウンロードしますが、ルートディレクトリ下のdocディレクトリはスキップします

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

- 操作事例 – 現在のバケットのルートディレクトリにあるすべてのファイルを上書きしてダウンロードします

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

> ⚠ 注意：
> ローカルに同名ファイルがある場合、ダウンロードは失敗しますので、 `-f` パラメータを使用してローカルファイルを上書きする必要があります。

- 操作事例 – 現在のbucketのルートディレクトリにあるすべてのファイルを同時にダウンロードし、同名ファイルのmd5チェックはスキップします

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

> ⚠ 注意：
> パラメータ `-s` または `--sync` を使用すると、フォルダをダウンロードする際に、すでにローカルに存在する同一ファイルをスキップすることができます（ただし、ダウンロードするファイルが

> COSCMDのuploadインターフェース経由でアップロードされたもので、そのファイルに `x-cos-meta-md5` ヘッダーがふくまれることが前提条件です）。

- 操作事例 – 現在のbucketのルートディレクトリにあるすべてのファイルを同時にダウンロードし、同サイズ・同名のファイルをスキップします

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

- 操作事例 – 現在のバケットのルートディレクトリにあるすべてのファイルを同時にダウンロードし、「クラウドでは削除されたがローカルでは削除されていないファイル」を同時に削除します

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

- 操作事例 – .txtと.docという拡張子を持つファイルを無視します

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

- 操作事例 – .txt拡張子を持つファイルを無視します

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

> ⚠ 注意：
> - フォルダをアップロードするときに、 `--ignore` パラメータを使用すると、いずれかのタイプのファイルを無視することができます。 `--include` パラメータを使用すると、いずれかのタイプのファイルをフィルタリングすることができます。shellワイルドカードルールや複数のルールをサポートし、カンマ `,` で区切ります。特定のサフィックスを無視する場合は、必ず `""` で囲む必要があります。
>   - `--ignore` を使用して特定のディレクトリ内のすべてのファイルをフィルタリングしたい場合は、絶対パスを使用し、パスの前後に `""` を追加する必要があります。例えば、 `coscmd upload -rs D:/doc / --ignore "D:/doc/ignore_folder/*"` などとします。

- 操作事例 – .txtと.docという拡張子を持つファイルをフィルタリングします

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

- 操作事例 – .txt拡張子を持つファイルをフィルタリングします

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

> ⚠ 注意：
> 古いバージョンのmgetインターフェースは廃止されました。downloadインターフェースはチャンク化されたダウンロードを使いますので、downloadインターフェースを使用してください。

## 署名入りダウンロードURLの取得

- コマンド形式

```
coscmd signurl <cospath>
```

- 操作事例 – doc/picture.jpgパスの署名付きURLを発行します

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

- 操作事例 – doc/picture.jpgパスの100s署名付きURLを発行します

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

> ⓘ 説明：
> - 「<>」のパラメータを、ダウンロードURLを取得する必要のあるCOS上のファイルのパス(cospath)に置き換えてください。
> - `-t time` を使用して、このURLの署名の有効期間（単位は秒）を設定します。デフォルトは10000sです。

## ファイルまたはフォルダの削除

### ファイル削除のコマンド形式

```
coscmd delete <cospath>
```

> ⚠ 注意：
> 「<>」内のパラメータを、削除する必要のあるCOS上のファイルのパス(cospath)に置き換えてください。ツールは、削除操作を確認するようユーザーに促します。

- 操作事例 – doc/exampleobject.txtを削除します

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

- 操作事例 – バージョンID付きのファイルを削除します

```
coscmd delete doc/exampleobject.txt --versionId
MTg0NDUxMzc4ODA3NTgyMTErEWN
```

### フォルダ削除のコマンド形式

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

- 操作事例 – docディレクトリを削除します

```
coscmd delete -r doc
```

- 操作事例 – folder/docディレクトリを削除します

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

- 操作事例 – docフォルダ下のすべてのバージョン管理ファイルを削除します

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

> **説明:**
> - 一括削除では `y` を入力して確定する必要がありますが、 `-f` パラメータを使用すると確認をスキップして直接削除することができます。
> - フォルダを削除するコマンドを実行すると、現在のフォルダとそのファイルが削除されますのでご注意ください。ただし、バージョン管理されたファイルを削除する場合は、バージョンIDを指定して削除する必要があります。

## マルチパートアップロードファイルのフラグメントの照会

- コマンド形式

```
coscmd listparts <cospath>
```

- 操作事例 – doc/プレフィックス付きファイルフラグメントを照会します

```
coscmd listparts doc/
```

## マルチパートアップロードファイルのフラグメントをクリア

- コマンド形式

```
coscmd abort
```

- 操作事例 – アップロードされたすべてのファイルフラグメントを削除します

```
coscmd abort
```

## ファイルまたはフォルダのコピー

### ファイルコピーのコマンド形式

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

- 操作事例 – 同じバケット内でコピーする場合、examplebucket-1250000000バケット内のpicture.jpgファイルをdocフォルダにコピーします

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

- 操作事例 – 異なるバケット内でコピーする場合、examplebucket2-1250000000バケットのdoc/picture.jpgオブジェクトをexamplebucket1-1250000000バケットのdoc/examplefolder/にコピーします

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

- ストレージタイプを変更し、ファイルタイプを低頻度ストレージに変更します

```
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'}"
```

- ストレージタイプを変更し、ファイルタイプをCASに変更して、photo.jpgとリネームします

```
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'}"
```

### フォルダコピーのコマンド形式

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

- 操作事例 – examplebucket2–1250000000バケット内のexamplefolderディレクトリをexamplebucket1–1250000000バケットのdocディレクトリにコピーします

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

**説明:**

- 「<>」内のパラメータを、コピーする必要のあるCOS上のファイルのパス(sourcepath)と、COS上にコピーする必要のあるファイルのパス(cospath)に置き換えてください。
- sourcepathの形式は、`<BucketName-APPID>.cos.<region>.myqcloud.com/<cospath>` です。
- –dパラメータを使用して `x-cos-metadata-directive` パラメータを設定します。オプション値はCopyとReplacedで、デフォルトはCopyです。
- –Hパラメータを使用してHTTP headerを設定する場合は、形式がJSONであることを確認してください。例:
  `coscmd copy -H -d Replaced "{'x-cos-storage-class':'Archive','Content-Language':'zh-CN'}" <localpath> <cospath>`
  。その他のヘッダーについては、PUT Object – Copyのドキュメントをご参照ください。

### ファイルまたはフォルダの移動

**注意:**

移動コマンドの
はと同一にすることはできません。同一にした場合はファイルが削除されます。`move`コマンドは先にコピーを行ってから削除するため、
`パスのファイルが最終的に削除されることが原因です。

### ファイル移動のコマンド形式

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

- 操作事例 – 同じバケット内で移動する場合、examplebucket-1250000000バケット内のpicture.jpgファイルをdocフォルダに移動します

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

- 操作事例 – 異なるバケット内で移動する場合、examplebucket2-1250000000バケット内のdoc/picture.jpgオブジェクトをexamplebucket1-1250000000バケットのdoc/folder/に移動します

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

- 操作事例 – ストレージタイプを変更し、ファイルタイプを低頻度ストレージに変更します

```
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'}"
```

- 操作事例 – ストレージタイプを変更し、ファイルタイプをCASに変更します

```
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'}"
```

### フォルダ移動のコマンド形式

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

- 操作事例 – examplebucket2–1250000000バケット内のexamplefolderディレクトリをexamplebucket1–1250000000バケットのdocディレクトリに移動します

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

**説明:**

- 「<>」内のパラメータを、移動させる必要のあるCOS上のファイルのパス(sourcepath)と、COSに移動する必要のあるファイルのパス(cospath)に置き換えてください。
- sourcepathの形式は、`<BucketName-APPID>.cos.<region>.myqcloud.com/<cospath>` です。
- –dパラメータを使用して `x-cos-metadata-directive` パラメータを設定します。オプション値はCopyとReplacedで、デフォルトはCopyです。
- –Hパラメータを使用してHTTP headerを設定する場合は、形式がJSONであることを確認してください。例:
  `coscmd move -H -d Replaced "{'x-cos-storage-class':'Archive','Content-Language':'zh-CN'}" <localpath> <cospath>`
  。その他のヘッダーについては、PUT Object – copy のドキュメントをご参照ください。

## オブジェクトのアクセス権限の設定

- コマンド形式

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

- 操作事例 – アカウント100000000001にpicture.jpgの読み込み権限を付与します

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

- 操作事例 – ファイルのアクセス権限を照会します

```
coscmd getobjectacl picture.jpg
```

## バージョン管理の有効化/一時停止

- コマンド形式

```
coscmd putbucketversioning <status>
```

- 操作事例 – バージョン管理を有効化します

```
coscmd putbucketversioning Enabled
```

- 操作事例 – バージョン管理を一時停止します

```
coscmd putbucketversioning Suspended
```

- 操作事例 – バージョン管理を照会します

```
coscmd getbucketversioning
```

⚠ 注意：
- 「<>」のパラメータを、ご希望のバージョン管理ステータス(status)に置き換えてください。
- 一度バージョン管理を有効にしたバケットは、バージョン管理を有効にしていない状態（初期ステータス）に戻すことはできません。ただし、このバケットのバージョン管理を一時停止することはでき、その後にアップロードされるオブジェクトでは複数のバージョンが生成されなくなります。

## アーカイブファイルのリカバリ

- アーカイブファイルリカバリのコマンド形式

```
coscmd restore <cospath>
```

- 操作事例 – 高速取得モードでpicture.jpgをリトリーブします。有効期間は3日間です

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

- アーカイブファイル一括リカバリのコマンド形式

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

- 操作事例 – 高速取得モードでexamplefolder/ディレクトリをリトリーブします。有効期間は3日間です

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

**説明:**

- 「<>」のパラメータを、ファイルリストを照会する必要のあるCOS上のファイルのパス(cospath)に置き換えてください。
- `-d <day>` を使用して、一時コピーの有効期限を設定します。デフォルト値は7です。
- `-t <tier>` を使用してリカバリモードを指定します。列挙値はExpedited （高速取得モード）、Standard （標準取得モード）、Bulk（一括取得モード）であり、デフォルト値はStandardです。

## よくあるご質問

COSCMDツールの使用に関するご質問は、COSCMDツールに関するよくあるご質問 をご参照ください。

# COS Migrationツール

最終更新日：：2024-06-26 10:27:35

## 機能説明

COS Migrationは、COSのデータ移行機能を統合したオールインワンツールです。簡単な設定操作により、ユーザーはローカルデータをCOSにすばやく移行することができ、以下のような特徴があります：

- 中断からの再開：ツールはアップロード時の中断からの再開をサポートしています。大容量ファイルについては、途中で終了した場合やサービスの障害があった場合、ツールを再実行して、アップロードが完了していないファイルのアップロードを再開することができます。
- マルチパートアップロード：COSにオブジェクトをチャンクでアップロードします。
- 並列アップロード：複数のオブジェクトの同時アップロードをサポートします。

> ⚠ 注意：
> - COS Migrationのエンコード形式は、UTF-8形式のみをサポートします。
> - このツールを使って同名ファイルをアップロードすると、デフォルトでは古い方の同名ファイルが上書きされます。同名ファイルをスキップするには、追加の設定が必要です。
> - ローカルデータ移行以外の場合、Migration Service Platformを優先して使用してください。
> - COS Migrationは**1回限り**の移行に用いるサービスであり、継続的な同期のシナリオには適しません。例えばローカルに毎日ファイルが追加され、COSに継続的に同期する必要がある場合、COS Migrationは移行タスクの重複を避けるために移行成功の記録を保存するため、継続的同期後の記録のスキャン時間が増大し続けることになります。このようなシナリオではファイルの同期の使用をお勧めします。

## 使用環境

### システム環境

WindowsとLinuxシステム。

### ソフトウェア依存

JDK 1.8 X64以降、JDKのインストールと設定の詳細については、Javaのインストールと設定をご参照ください。

- Linux環境ではIFUNCのサポートが必要です。環境のbinutilsのバージョンをチェックし、2.20以降のバージョンであることを確認します。

## 利用方法

### 1. ツールの取得

COS Migrationツール に移動してダウンロードします。

## 2. 解凍ツールキット

### Windows

解凍して、次のようにディレクトリに保存します。

```
C:\Users\Administrator\Downloads\cos_migrate
```

### Linux

解凍して、いずれかのディレクトリに保存します。

```
unzip cos_migrate_tool_v5-master.zip && cd cos_migrate_tool_v5-master
```

### マイグレーションツールの構造

COS Migrationツールを正しく解凍すると、次のようなディレクトリ構造になります。

```
COS_Migrate_tool
|--conf  #設定ファイルが配置されているディレクトリ
|   |--config.ini  #設定ファイルの移行
|--db    #移行に成功した記録を保存
|--dep   #プログラムのメインロジックコンパイルによって発行されたJARパッケージ
|--log   #ツールの実行中に発行されたログ
|--opbin #コンパイル用スクリプト
|--result #マイグレーション成功記録を保存するためのディレクトリであり、記録のファイル名は 「期日.out」、形式は「絶対パス¥ｔファイルサイズ¥t最終変更時刻」
|--src   #ツールのソースコード
|--tmp   #一時ファイルストレージディレクトリ
|--.gitignore   #git バージョン管理で無視されたファイルとフォルダ
|--pom.xml #プロジェクト設定ファイル
|--README  #説明ドキュメント
|--start_migrate.sh  #Linuxでの移行起動スクリプト
|--start_migrate.bat #Windowsでの移行起動スクリプトdbディレクトリには、主にツールの移行に成功したファイル識別子が記録されています。各移行タスクはdb内の記録に優先順位をつけ、現在のファイル識別子がすでに記録されている場合は現在のファイルをスキップし、それ以外の場合はファイルの移行を実行します。
```

- ログディレクトリには、ツールの移行に伴うすべてのログが記録されています。移行中にエラーが発生した場

合は、まずこのディレクトリ内のerror.logをご確認ください。

> **説明:**
> - dbディレクトリには、主にツールの移行に成功したファイル識別子が記録されています。各移行タスクはdb内の記録に優先順位をつけ、現在のファイル識別子がすでに記録されている場合は現在のファイルをスキップし、それ以外の場合はファイルの移行を実行します。
> - ログディレクトリには、ツールの移行に伴うすべてのログが記録されています。移行中にエラーが発生した場合は、まずこのディレクトリ内のerror.logをご確認ください。

## 3. 設定ファイルconfig.iniの変更

移行起動スクリプトを実行する前に、設定ファイルconfig.iniを変更する必要があります（パス: `./conf/config.ini` ）。config.iniの内容は、以下のセクションに分けられます。

### 3.1 移行タイプの設定

type はマイグレーションのタイプを示し、固定入力とします `type=migrateLocal` 。

```
[migrateType]
type=migrateLocal
```

### 3.2 移行タスクの設定

ユーザーは、主にターゲットCOS情報への移行の設定や移行タスク関連の設定など、実際の移行要件に基づいた設定を行います。

```
# マイグレーションツールのパブリック設定セクションには、ターゲットcosに移行する必要のあるアカウント情報が含まれます。
[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
executeTimeWindow=00:00,24:00
outputFinishedFileFolder=./result
resume=false
skipSamePath=false
requestTryCount=5
```

| 名称 | 説明 | デフォルト値 |
| --- | --- | --- |
| secretId | ユーザーキーのSecretIdです。`COS_SECRETID`を実際のキー情報に置き換えてください。CAMコンソールのTencent Cloud APIキー画面に進むと取得できます | - |
| secretKey | ユーザーキーのSecretKeyです。`COS_SECRETKEY`を実際のキー情報に置き換えてください。CAMコンソールのTencent Cloud APIキー画面に進むと取得できます | - |
| bucketName | ターゲットBucketの名称で、命名形式は`<BucketName-APPID>`。Bucket名には必ずAPPIDを含める必要があります。例：examplebucket-1250000000 | - |
| region | ターゲットBucketのRegion情報です。COSのリージョンの略称については、リージョンとアクセスドメイン名をご参照ください | - |
| storageClass | データ移行後のストレージタイプです。オプション値はStandard（標準ストレージ）、Standard_IA（低頻度ストレージ）、Archive(アーカイブストレージ)、Maz_Standard（複数のAZを備えた標準ストレージ）、Maz_Standard_IA（複数のAZを備えた低頻度ストレージ）です。詳細については、ストレージタイプの概要をご参照ください | Standard |
| cosPath | 移行先のCOSパスです。`/`はBucketのルートパスへの移行、`/folder/doc/`はBucketの`/folder/doc/`への移行を表します。`/folder/doc/`が存在しない場合は、自動的にパスが作成されます | / |
| https | HTTPS通信を使用するかどうかです。onはオン、offはオフを表します。オンの場合は通信速度が遅くなり、セキュリティ要件の高いシナリオに適しています | off |
| tmpFolder | 他のクラウドストレージからCOSへの移行中に一時ファイルを保存するために使用されたディレクトリは、移行が完了すると削除されます。形式は絶対パスである必要があります。<br>Linuxの場合、`/a/b/c`のようにシングルスラッシュで区切ります。 | ./tmp |

| | | |
|---|---|---|
| | Windowsの場合、`E:\\a\\b\\c`のようにダブルバックスラッシュ区切ります。<br>デフォルトでは、ツールが配置されているパスのtmpディレクトリです | |
| smallFileThreshold | 小さなファイルの閾値のバイトです。この閾値以上であれば、マルチパートアップロードを使用します。それ以外の場合はシンプルアップロードを使用します。デフォルトは5MB（5242880 Byte）です | 5242880 |
| smallFileExecutorNum | 小さなファイル（smallFileThresholdより小さいファイル）の同時実行性で、シンプルアップロードを使用します。パブリックネットワーク経由でCOSに接続しており、帯域幅が小さい場合は、この同時実行性を低下させてください | 64 |
| bigFileExecutorNum | 大きなファイル（smallFileThreshold以上のファイル）の同時実行性で、マルチパートアップロードを使用します。パブリックネットワーク経由でCOSに接続しており、帯域幅が小さい場合は、この同時実行性を低下させてください | 8 |
| entireFileMd5Attached | マイグレーションツールがフルテキストのMD5を計算し、その後の検証のためにファイルのカスタムヘッダーx-cos-meta-md5に格納することを表します。COSのマルチパートアップロードされた大きなファイルのetagは、フルテキストのMD5ではないからです | on |
| executeTimeWindow | 実行時間ウィンドウ。時刻粒度は分単位です。このパラメータは、マイグレーションツールが毎日実行される時間帯を定義します。例えば:<br>パラメータ 03:30,21:00は、タスクが朝 03:30 から夜 21:00 の間に実行され、それ以外の時間はスリープ状態に入ることを意味します。スリープ状態でマイグレーションを一時停止すると、マイグレーションの進行状況が維持され、次のタイムウィンドウになると、自動的に続行されます。後の時点が前の時点より大きい必要があることに注意してください。 | 00:00,24:00 |
| outputFinishedFileFolder | このディレクトリには、移行に成功した結果が保存されます。結果のファイルは、例えば`./result/2021-05-27.out`のように日付に基づいて命名されます。ここで./resultは、作成済みのディレクトリです。ファイル内容の各行の形式は、絶対パス\tファイルサイズ\t最終更新時刻です。設定を空欄にすると、結果は出力されません。 | ./result |
| resume | ソースファイルのリストを、最後に実行した結果までトラバーサル処理し続けるかどうかです。デフォルトは先頭からです。 | false |
| skipSamePath | COSにすでに同名ファイルがある場合に、そのままスキップするかどうかです。デフォルトではスキップされず、元のファイルが上書きされます。 | false |

| | | |
|---|---|---|
| requestTryCount | 各ファイルのアップロードの合計試行回数。 | 5 |

### 3.3 データソース情報の設定

#### 3.3.1 ローカルのデータソースmigrateLocalの設定

ローカルからCOSに移行する場合は、この部分の設定を行います。具体的な設定項目および説明は以下のとおりです。

```
# ローカルからcosへの移行の設定セクション
[migrateLocal]
localPath=E:\\code\\java\\workspace\\cos_migrate_tool\\test_data
excludes=
ignoreModifiedTimeLessThanSeconds=
```

| 設定項目 | 説明 |
|---|---|
| localPath | ローカルディレクトリ。形式は絶対パスである必要があります。<br>• Linuxの場合、 `/a/b/c` のようにシングルスラッシュで区切ります。<br>• Windowsの場合、 `E:\\a\\b\\c` のようにダブルバックスラッシュで区切ります<br>注意：このパラメータはディレクトリのパスのみを入力でき、具体的なファイルのパスは入力できません。具体的なファイルのパスを入力した場合は目的のオブジェクト名の解析エラーとなり、cosPath=/の場合、バケット作成リクエストであると誤って解析されます |
| excludes | 除外するディレクトリまたはファイルの絶対パスで、localPathより下のディレクトリまたはファイルは移行されないことを表します。複数の絶対パスはセミコロンで区切ります。空欄の場合、localPathより下のすべてが移行されることを表します |
| ignoreModifiedTimeLessThanSeconds | 更新時間が現在時刻から一定時間に満たないファイルを除外します。単位は秒で、デフォルトでは設定されていません。これは、lastmodified時刻に基づくフィルタリングを行わないことを表します。ファイルの更新と同時にマイグレーションツールを実行し、更新中のファイルがCOSに移行、アップロードされないようにしたいお客様向けです。例えば300に設定すると、更新から5分以上経過したファイルのみをアップロードします |

## 4. マイグレーションツールの実行

### Windows

**start_migrate.bat**をダブルクリックして実行します。

### Linux

1.設定ファイルconfig.iniから設定を読み込み、次のコマンドを実行します。

```
sh start_migrate.sh
```

2.一部のパラメータは、コマンドラインから設定を読み込み、次のコマンドを実行します。

```
sh start_migrate.sh -Dcommon.cosPath=/savepoint0403_10/
```

**説明:**

- ツールは、コマンドラインからの読み取りと設定ファイルからの読み取りという2種類の設定項目の読み取りをサポートしています。
- コマンドラインは設定ファイルより優先されます。同じ設定オプションであれば、コマンドラインのパラメータが優先されるということです。
- コマンドラインで設定項目を読み取る形式により、Bucket名、COSパス、移行するソースパスなど、2つのタスクの主要な設定項目が完全に同一でない場合、ユーザーは異なる移行タスクを同時に実行しやすくなります。さまざまな移行タスクを異なるdbディレクトリに書き込むことで、同時移行が確保されるからです。上記のツール構造のdb情報をご参照ください。
- 設定項目の形式は、**–D{sectionName}.{sectionKey}={sectionValue}**です。ここで、**sectionNameは設定ファイルのセクション名、sectionKeyはセクション内の設定項目名、sectionValueはセクション内の設定項目値を表します。移行先のCOSのパスを設定した場合、**–Dcommon.cosPath=/bbb/ddd**で表されます。

# 移行のメカニズムとプロセス

## 移行メカニズムの原理

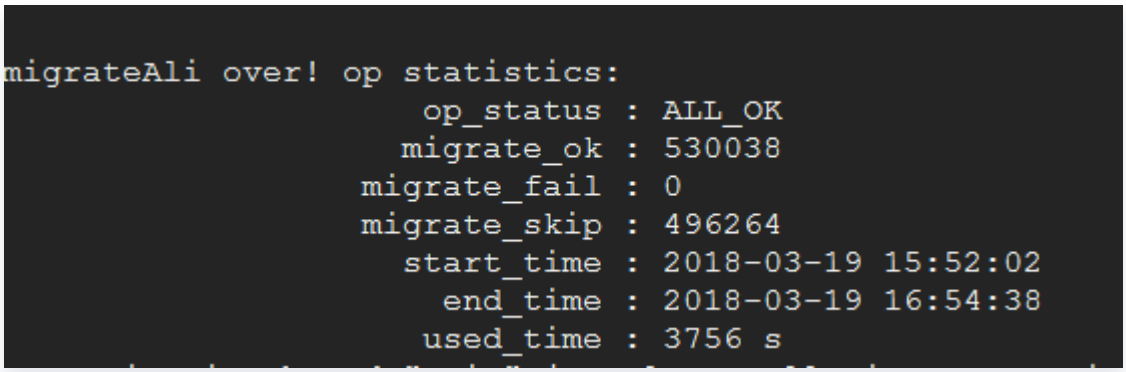
COS マイグレーションツールには状態があり、マイグレーションに成功したものは db ディレクトリに記録され、KV の形として leveldb ファイルに保存されます。毎回マイグレーションする前に、マイグレーションするパスが db に存在するかどうかをチェックします。もし存在し、且つ mtime が db に存在するものと一致している場合にはマイグレーションをスキップし、そうしない場合にはマイグレーションします。そのため、COS を探すのではなく、db の中にマイグレーション成功の記録があるかどうかを確認するわけです。マイグレーションツールの代わりに、他の方法（COSCMD やコンソールなど）を通じてファイルを削除・変更した場合、マイグレーションツールを実行しても、その変更を察知しないため、再度マイグレーションを行わないことになります。

## 移行プロセスの手順

1. 設定ファイルが読み取られ、セクションがマイグレーションtypeに従って読み取られ、パラメータがチェックされます。
2. db においてマイグレーションするファイルのマークをスキャンして比較し、アップロードが許可されている

かどうかを判断します。

3. マイグレーション実行中、実行結果がプリントアウトされます。そのうち、inprogress はマイグレーション中、skip はスキップ、fail は失敗、ok は成功、condition_not_match はマイグレーション条件を満たさないためにスキップされたファイル（lastmodifed や excludes など）を示します。失敗の詳細情報は log の error ログで確認することができます。
4. 移行全体の終了時に、累積した移行の成功数、失敗数、スキップ数、所要時間などの統計情報が出力されます。マイグレーションツールは成功したものをスキップして失敗したものを再移行するので、失敗した場合はerrorログを確認するか、再実行してください。実行完了結果の略図は以下のとおりです。




## よくあるご質問

COS Migrationツールの使用中に移行失敗や実行エラーといった異常が発生した場合は、COS Migrationツールに関するよくあるご質問をご参照ください。

## 結論

もちろん、COS は上記のアプリケーションとサービスを提供するだけでなく、多くの人気のあるオープンソースアプリケーションをも提供し、Tencent Cloud COS プラグインを統合しております。 こちら をクリックしてワンクリックで起動し、すぐにお使いになることを歓迎します！

# FTP Serverツール

最終更新日：：2025-11-14 18:06:20

## 機能の説明

COS FTP Serverは、FTPプロトコルによるCloud Object Storage（COS）内のオブジェクトやディレクトリの直接操作をサポートしています。これにはファイルのアップロード、ダウンロード、削除、フォルダ作成などが含まれます。FTP Serverツールは Pythonで実装されており、インストールがより簡単です。

## 機能説明

**アップロードメカニズム**：ストリーミングアップロードはローカルディスクに記録されることなく、標準的なFTPプロトコルに従って作業ディレクトリを設定するだけで、実際のディスクストレージ容量を占有することはありません。
**ダウンロードメカニズム**：ダイレクトストリーミングでクライアントへ戻ります。
**ディレクトリメカニズム**：bucketはFTP Server全体のルートディレクトリであり、bucketの下に複数のサブディレクトリを作成することができます。
**複数bucketのバインド**：複数のbucketの同時バインドをサポートします。

> **説明：**
> 複数bucketのバインド：異なるFTP Serverの作業パス(home_dir)を介して実装されるため、異なるbucketとユーザー情報を指定する場合は、home_dirが異なることを確認する必要があります。

**削除操作の制限**：新しいFTP Serverでは、各ftpユーザーに対しdelete_enableオプションを設定して、そのFTPユーザーがファイルの削除を許可されているかどうかを識別できるようになっています。
**サポートされているFTPコマンド：** put、mput、get、rename、delete、mkdir、ls、cd、bye、quite、size。
**サポートされていないFTPコマンド：** append、mget（ネイティブのmgetコマンドはサポートされていませんが、FileZillaクライアントなど一部のWindowsクライアントでは、引き続き一括ダウンロードが可能です）

> **説明：**
> FTP Serverツールは現在、中断からの再開機能をサポートしていません。

## 使用の開始

### システム環境

- オペレーティングシステム：Linux。Tencent CentOSシリーズCVMを推奨します。現時点ではWindowsシステムはサポートしていません。
- psutilが依存するLinuxシステムパッケージ：python-devel（またはpython-dev。Linuxディストリビューションによって名前が異なります）。`yum install python-devel` や

`aptitude install python-dev` などのLinuxのパッケージ管理ツールによって追加されます。

- Pythonインタープリターバージョン：Python 2.7、インストールと設定は、Pythonのインストールと設定 を参照して行ってください。

> **説明：**
> FTP Serverツールは、Python 3をサポートしていません。

- 依存パッケージ：
- cos-python-sdk-v5 （≥1.6.5）
- pyftpdlib （≥1.5.2）
- psutil (>=5.6.1)

### 使用制限

COS XMLバージョンに適用します。

### インストールと実行

FTP Serverツールのダウンロードアドレスは、cos-ftp-server です。インストール手順は次のとおりです。

1. FTP Serverのディレクトリに移動してsetup.pyを実行し、FTP Serverとそれに関連する依存ライブラリをインストールします（インターネットへのアクセスが必要です）。

```
python setup.py install    # ここでは、お客様のアカウントのsudoまたはroot権限が必要な場合があります。
```

2. サンプル設定ファイル `conf/vsftpd.conf.example` をコピーし、 `conf/vsftpd.conf` というファイル名にします。本ドキュメントの 設定ファイル の項を参照し、bucketやユーザー情報を適切に設定します。
3. ftp_server.pyを実行して、FTP Serverを起動します。

```
python ftp_server.py
```

また、FTP Serverの起動方法には、次の2通りがあります。

- nohupコマンドを使用して、バックグラウンドプロセスとして起動します。

```
nohup python ftp_server.py >> /dev/null 2>&1 &
```

- screenコマンドを使用して、バックグラウンドで実行します（screenツールをインストールする必要があります）。

```
screen -dmS ftp
screen -r ftp
python ftp_server.py
#ショートカットキーCtrl+A+Dを使用して、メインscreenに戻ります。
```

### 実行の停止

- FTP Serverを直接起動する場合、またはFTP Serverをバックグラウンドでscreenモードで実行する場合は、ショートカットキー `Ctrl+C` を使用すると、FTP Serverの実行を停止することができます。
- nohupコマンドで起動した場合は、以下のように停止させることができます。

```
ps -ef | grep python | grep ftp_server.py | grep -v grep | awk
'{print $2}' | xargs -I{} kill {}
```

## 設定ファイル

FTP Serverツールのサンプル設定ファイルは、 `conf/vsftpd.conf.example` ですので、これをコピーしてvsftpd.confという名前を付け、次の設定項目に従って設定してください。

```
[COS_ACCOUNT_0]
cos_secretid = COS_SECRETID     # お客様のSECRETIDに置き換えてください
cos_secretkey = COS_SECRETKEY   # お客様のSECRETKEYに置き換えてください
cos_bucket = examplebucket-1250000000
cos_region = region    # お客様のバケットリージョンに置き換えてください
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user0       # FTPをマウントしたいローカルパスに置き換えてくださ
い（マシン上に実際に存在するパスに設定する必要があります。ソフトリンクはサポートしてい
ません）
ftp_login_user_name=user0    # ユーザー定義のアカウントに置き換えてください
ftp_login_user_password=pass0    # ユーザー定義のパスワードに置き換えてください
authority=RW                 # このユーザーの読み書き権限を設定します。Rは読み込
み、Wは書き込み、RWは読み書き両方の権限を意味します
delete_enable=true        # このftpユーザーの削除を許可する場合はtrue（デフォル
ト）、禁止する場合はfalseとします

[COS_ACCOUNT_1]
cos_secretid = COS_SECRETID     # お客様のSECRETIDに置き換えてください
cos_secretkey = COS_SECRETKEY   # お客様のSECRETKEYに置き換えてください
```

```
cos_bucket = examplebucket-1250000000
cos_region = region    # お客様のバケットリージョンに置き換えてください
cos_protocol = https
#cos_endpoint = region.myqcloud.com
home_dir = /home/user1       # FTPをマウントしたいローカルパスに置き換えてください（マシン上に実際に存在するパスに設定する必要があります。ソフトリンクはサポートしていません）
ftp_login_user_name=user1   # ユーザー定義のアカウントに置き換えてください
ftp_login_user_password=pass1    #ユーザー定義のパスワードに置き換えてください
authority=RW                # このユーザーの読み書き権限を設定します。Rは読み込み、Wは書き込み、RWは読み書き両方の権限を意味します
delete_enable=false       # このftpユーザーの削除を許可する場合はtrue（デフォルト）、禁止する場合はfalseとします

[NETWORK]
# FTP ServerがゲートウェイまたはNATの背後にある場合は、この設定項目を使用してゲートウェイのIPアドレスまたはドメイン名をFTPに割り当てることができます
masquerade_address = XXX.XXX.XXX.XXX
# FTP Serverのリスニングポート、デフォルトは2121です。ファイアウォールがこのポートを開放する必要があるので、ご注意ください（例えば、FTP ServerツールをTencent Cloud CVMにデプロイする場合は、CVMセキュリティグループでこのポートを開放する必要があります）
listen_port = 2121
# passive_portは、passiveモードでポートの選択範囲を設定できます。デフォルトでは、[60000, 65535]の範囲で選択されます。ファイアウォール（CVMセキュリティグループなど）は、この範囲のポートを開放する必要があるので、ご注意ください
passive_port = 60000,65535


[FILE_OPTION]
# デフォルトの単一ファイルサイズは最大200Gをサポートしますが、大きすぎるサイズを設定することはお勧めしません
single_file_max_size = 21474836480

[OPTIONAL]
# 以下の設定は、特に必要がなければdefaultのままにしておくことをお勧めします。設定する必要がある場合は、適切な整数を入力してください
min_part_size        = default
upload_thread_num    = default
max_connection_num   = 512
```

```
max_list_file        = 10000                # lsコマンドでリストアップできる
ファイルの最大数で、大きすぎない値に設定することをお勧めします。大きすぎる場合、lsコマ
ンドの遅延が長くなります
log_level            = INFO                 # ログ出力レベルを設定します
log_dir              = log                  # ログをストレージするディレクトリ
を設定します。デフォルトは、FTP Serverディレクトリの下のログディレクトリです
```

**説明:**

- 各ユーザーを異なるbucketにバインドする場合は、[COS_ACCOUNT_X]のsectionを追加するだけで可能です。
  それぞれのCOS_ACCOUNT_Xのsectionには、次のような説明があります
- 各ACCOUNTのユーザー名(ftp_login_user_name)とユーザーのホームディレクトリ(home_dir)は、それぞれ異なっていなければなりません。また、ホームディレクトリは、システム上の実際のディレクトリである必要があります。
- 各COS FTP Serverに同時にログインできるユーザー数は100以下とします。
- endpointとregionは同時に有効になりません。パブリッククラウドCOSサービスを使用するには、regionフィールドを正しく入力するだけで可能です。endpointは一般的に、プライベートなデプロイ環境で使用されます。regionとendpointの両方が入力された場合、endpointが優先して有効になります。
- 設定ファイルのOPTIONALオプションは、上級ユーザーがアップロード性能を調整するためのオプションです。マシンの性能に応じてアップロードスライスのサイズと同時アップロードするスレッド数を適度に調整することによって、アップロード速度を高めることができます。通常、ユーザーはこれを調整する必要はなく、デフォルト値のままで問題ありません。
  また、最大接続数を制限するオプションも提供しています。最大接続数を制限したくない場合は、ここに0を入力します。0は、最大接続数が制限されていないことを意味します（ただし、マシンの性能に応じて合理的に評価する必要があります）。
- 通常は、設定ファイルのmasquerade_address設定項目において、クライアントがCOS FTP Serverに接続するために使用するIPアドレスを指定することをお勧めします。この点に関してご質問がある場合は、FTP Serverツールのよくあるご質問のドキュメントをご参照ください。
- FTP Serverに複数のIPアドレスがある場合、ifconfigコマンドを実行すると、パブリックネットワークにマッピングされたネットワークカードIPは10.xxx.xxx.xxxとなり、マッピングしたパブリックネットワークIPは119.xxx.xxx.xxxとみなされます。この際、FTP Serverが明示的にmasquerade_addressをパブリックネットワークIP(119.xxx.xxx.xxx)として設定しない場合、FTP ServerはPassiveモードでプライベートネットワークアドレス(10.xxx.xxx.xxx)を使用してクライアントにパケットを返すことがあります。この場合、クライアントはFTP Serverに接続できますが、クライアントにパケットを正しく返すことができません。したがって通常は、クライアントがServerに接続する際に使用するIPアドレスをmasquerade_addressに設定することをお勧めします。
- 設定ファイルのlisten_port設定項目は、COS FTP Serverのリスニングポートであり、デフォルトは

2121です。passive_port設定項目は、COS FTP Serverのデータチャネルリスニングポート範囲であり、デフォルトでは[60000, 65535]の範囲で選択されています。クライアントがCOS FTP Serverに接続するときは、ファイアウォールがlisten_portとpassive_portに設定されたポートを開放していることを確認する必要があります。

# クイックプラクティス

## Linuxftpコマンドを使用したCOS FTP Serverへのアクセス

1. Linux ftpクライアントをインストールします。

```
yum install -y ftp
```

2. Linuxコマンドラインでコマンド `ftp [ipアドレス] [ポート番号]` を使用して、COS FTP Serverに接続します。例えば、次のコマンドです。

```
ftp 192.xxx.xx.103 2121
```

   - ftpコマンドでは、IP設定はサンプル設定ファイル `conf/vsftpd.conf.example` の **masquerade_address**設定項目に対応しています。この例では、IPは192.xxx.xx.103に設定されています。
   - ftpコマンドでは、ポート設定はサンプル設定ファイル `conf/vsftpd.conf.example` の**listen_port**設定項目に対応しています。この例では、2121に設定されています。

3. 上記のコマンドを実行すると、**Name**と**Password**という入力すべき項目が表示されますので、COS FTP Serverの設定項目ftp_login_user_nameとftp_login_user_passwordに設定した内容を入力すると、接続が完了します。
   - **Name**：サンプル設定ファイル `conf/vsftpd.conf.example` の **ftp_login_user_name**設定項目に対応しています（設定が必要です）。
   - **Password**：サンプル設定ファイル `conf/vsftpd.conf.example` の **ftp_login_user_password**設定項目に対応しています（設定が必要です）。

## FileZillaコマンドを使用したCOS FTP Serverへのアクセス

1. FileZillaクライアント をダウンロードしてインストールします。
2. FileZillaクライアントでCOS FTP Serverのアクセス情報を設定した後、**クイック接続**をクリックします。
   - ホスト(H)：サンプル設定ファイル `conf/vsftpd.conf.example` の **masquerade_address**設定項目に対応しています。この例では、IPは192.xxx.xx.103に設定されています。

     ⚠ **注意：**

> COS FTP Serverがゲートウェイまたは NATの背後にある場合は、この設定項目を使用してゲートウェイのIPアドレスまたはドメイン名をCOS FTP Serverに割り当てることができます。

- **ユーザー名(U)**：サンプル設定ファイル `conf/vsftpd.conf.example` の **ftp_login_user_name**設定項目に対応しています（設定が必要です）。
- **パスワード(W)**：サンプル設定ファイル `conf/vsftpd.conf.example` の **ftp_login_user_password** 設定項目に対応しています（設定が必要です）。
- **ポート(P)**：サンプル設定ファイル `conf/vsftpd.conf.example` の**listen_port**設定項目に対応しています。この例では、2121に設定されています。

## よくあるご質問

FTP Serverツールの使用時にエラーが発生した場合やアップロードの制限についてご質問がある場合は、FTP Serverツールのよくあるご質問をご参照ください。

# Hadoopツール

最終更新日：：2025-10-15 16:48:08

## 機能説明

Hadoop-COSは、Tencent Cloud Object Storage(COS)をベースとして標準的なHadoopファイルシステムを実装しています。これにより、Hadoop、Spark、Tezなどのビッグデータコンピューティングフレームワークが COSと統合し、HDFSファイルシステムにアクセスする場合と同じように、COSに保存されたデータの読み書きを可能にするサポートを提供しています。Hadoop-COSは、URIのスキームとしてcosnを使用するため、CosNファイルシステムとも呼ばれます。

## 使用環境

### システム環境

Linux、Windows、macOSシステムがサポートされています。

### ソフトウェア依存

Hadoop-2.6.0およびそれ以降のバージョン。

> **説明：**
> - 現在、Hadoop-COSは、Apache Hadoop-3.3.0に正式に統合されました 公式統合 。
> - Apache Hadoop-3.3.0以前のバージョンまたはCDHがHadoop-cos jarパッケージを統合した後、NodeManagerを再起動してjarパッケージをロードする必要があります。
> - 特定のHadoopバージョンのjarパッケージをコンパイルする必要がある場合は、pomファイルのhadoop.versionを変更してコンパイルできます。

## ダウンロードとインストール

### Hadoop-COSディストリビューションパッケージとその依存関係の取得

ダウンロードアドレス：Hadoop-COS release 。

### Hadoop-COSプラグインのインストール

1. `hadoop-cos-{hadoop.version}-{version}.jar` と `cos_api-bundle-{version}.jar` を `$HADOOP_HOME/share/hadoop/tools/lib` にコピーします。

> **説明：**
> Hadoopの特定バージョンに応じて対応するjarパッケージを選択します。releaseの中に対応するバージョンのjarパッケージが提供されていない場合は、pomファイルのHadoopバージョン番号を変更す

ることで再コンパイルと発行ができます。

2. hadoop-env.shファイルを変更します。`$HADOOP_HOME/etc/hadoop` ディレクトリに移動し、hadoop-env.shファイルを編集して、以下の内容を追加し、cosn関連のjarパッケージをHadoop環境変数に追加します。

```
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
```

## 設定方法

### Hadoopの設定

`$HADOOP_HOME/etc/hadoop/core-site.xml` を変更して、COS関連のユーザーおよび実装クラスの情報を追加します。次に例を示します。

**注意：**
設定項目を変更する場合、COSNの `core-site.xml` ファイル内の設定項目を直接変更できます。Sparkを使用して動的にHadoop設定をロードする場合、設定項目の前に `spark.hadoop.**` というプレフィックスを追加する必要があります。例えば、 `spark.hadoop.fs.cosn.upload.buffer` です。この設定項目は、 `SparkContext` の `hadoopConfiguration` で設定するか、シェルで `spark-sql --conf` を通じて指定できます。

```
<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>xxxxxxxxxxxxxxxxxxxxxxxx</value>
        <description>Tencent Cloud Secret Id</description>
    </property>

    <property>
        <name>fs.cosn.userinfo.secretKey</name>
        <value>xxxxxxxxxxxxxxxxxxxxxxxx</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 filesysten, 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>
```

fs.defaultFSは本番環境での設定を推奨しません。一部のテストシーン（例：hive-testbenchなど）での使用が必要な場合でも、fs.defaultFSが1つだけ設定されていることを確認してください。以下の設定情報を追加できます：

```
<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>
```

## 設定項目の説明

| プロパティキー | 説明 | デフォルト値 | 入力必須項目 |
| --- | --- | --- | --- |
| fs.cosn.userinfo.secretId/secretKey | お客様のアカウントのAPIキー情報を入力します。CAMコンソールにログインすれば、Tencent Cloud APIキーを確認することができます。 | なし | はい |
| fs.cosn.credentials.provider | SecretIdとSecretKeyの取得方法を設定します。現在サポートされている方法は次のとおりです。<br>1. org.apache.hadoop.fs.auth.SessionCredential Provider：リクエストURIからsecret idとsecret keyを取得します。その形式は、次のとおりです。cosn://{secretId}:{secretKey}@examplebucket-1250000000/。<br>2. org.apache.hadoop.fs.auth.SimpleCredentialProvider：core-site.xml設定ファイルからfs.cosn.userinfo.secretIdとfs.cosn.userinfo.secretKeyを読み込み、SecretIdとSecretKeyを取得します。<br>3. org.apache.hadoop.fs.auth.EnvironmentVariableCredential Provider：システム環境変数のCOS_SECRET_IDとCOS_SECRET_KEYから取得します。<br>4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider：一時キー形式を使用してアクセスします。 | この設定項目が指定されていない場合、デフォルトで<br><br>以下の順序で読み込みます。<br>1. org.apache.hadoop.fs.auth.SessionCredentialProvider<br>2. org.apache.hadoop.fs.auth.SimpleCredentialProvider<br>3. org.apache.hadoop.fs.auth.EnvironmentVariableCredentialProvider<br>4. org.apache.hadoop.fs.auth.SessionTokenCredentialProvider<br>5. org.apache.hadoop.fs.auth.CVMInstanc | いいえ |

| | | | |
|---|---|---|---|
| | 5. org.apache.hadoop.fs.auth.CVMInstanceCredentialsProvider：Tencent CloudのCloud Virtual Machine(CVM)にバインドされたロールを使用して、COSにアクセスするための一時キーを取得します<br>6. org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider：Tencent CloudのCloud Physical Machine(CPM)にバインドされたロールを使用して、COSにアクセスするための一時キーを取得します。<br>7. org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider：Tencent Cloud EMRインスタンスにバインドされたロールを使用して、COSにアクセスするための一時キーを取得します。<br>8. org.apache.hadoop.fs.auth.RangerCredentialsProviderは、rangerを使用してキーを取得します。 | eCredentialsProvider<br>6. org.apache.hadoop.fs.auth.CPMInstanceCredentialsProvider<br>7. org.apache.hadoop.fs.auth.EMRInstanceCredentialsProvider | |
| fs.cosn.useHttps | COSバックエンドでのトランスポートプロトコルとしてHTTPSを使用するかどうかを設定します。 | true | いいえ |
| fs.cosn.impl | FileSystem用のcosn実装クラス、org.apache.hadoop.fs.CosFileSystemに固定されます。 | なし | はい |
| fs.AbstractFileSystem.cosn.impl | AbstractFileSystem用のcosn実装クラス、org.apache.hadoop.fs.CosNに固定されます。 | なし | はい |
| fs.cosn.bucket.region | アクセスするバケットのリージョン情報を入力してください。列挙値については、リージョンとアクセスドメイン名のリージョンの略称をご参照ください。<br>例えば、ap-beijing、ap-guangzhouなどです。元の設定：fs.cosn.userinfo.regionと互換性があります。 | なし | はい |
| fs.cosn.bucket.endpoint_suffix | 接続するCOS endpointを指定します。この項目は入力必須項目ではありません。パブリッククラウドCOSユーザーの場合、<br>上記のregionの設定を正しく入力するだけです。元の設定：<br>fs.cosn.userinfo.endpoint_suffixと互換性があります。この項目を有効にするために、設定時 | なし | いいえ |

| | | | |
|---|---|---|---|
| | にfs.cosn.bucket.regionの設定項目のendpointを削除してください。 | | |
| fs.cosn.tmp.dir | 実際に存在するローカルディレクトリを設定してください。プロセス実行中に生成された一時ファイルは、一時的にここに置かれます。 | /tmp/hadoop_cos | いいえ |
| fs.cosn.upload.part.size | CosNファイルシステムの各blockのサイズです。マルチパートアップロードされた各part sizeのサイズでもあります。COSはマルチパートアップロードで最大10,000ブロックまでしかサポートできないため、使用できる最大1ファイルサイズを推定する必要があります。<br>例えば、part sizeが8MBの場合、最大で78GBの単一ファイルのアップロードに対応します。<br>part sizeは最大2GBまで対応可能で、1ファイルあたり最大19TBまで対応できます。 | 8388608（8MB） | いいえ |
| fs.cosn.upload.buffer | CosNファイルシステムがアップロード時に依存するバッファのタイプです。現在、非ダイレクトメモリバッファ(non_direct_memory)<br>ダイレクトメモリバッファ(direct_memory)、ディスクマップバッファ(mapped_disk)という3種類のバッファがサポートされています。非ダイレクトメモリバッファ<br>エリアではJVMヒープメモリが使用されますが、ダイレクトメモリバッファエリアではオフヒープメモリが使用されます。ディスクマップバッファはメモリファイルマッピングをベースとするバッファエリアです。 | mapped_disk | いいえ |
| fs.cosn.upload.buffer.size | CosNファイルシステムがアップロードに依存するバッファのサイズです。-1に指定されている場合、バッファに制限がないことを意味します。<br>バッファサイズが制限されていない場合、バッファタイプはmapped_diskである必要があります。指定されたサイズが0より大きい場合、この値は少なくとも1つのblockのサイズ以上である必要があります。 元の設定:<br>fs.cosn.buffer.sizeと互換性があります。 | -1 | いいえ |
| fs.cosn.block.size | CosNファイルシステムのblock sizeです。 | 134217728(128MB) | いいえ |
| | | | |

| | | | |
|---|---|---|---|
| fs.cosn.upload_thread_pool | ファイルがCOSにストリーミングされるときの同時アップロードスレッドの数です。 | 10 | いいえ |
| fs.cosn.copy_thread_pool | ディレクトリのコピー操作中にファイルを同時にコピーおよび削除するために使えるスレッドの数です。 | 3 | いいえ |
| fs.cosn.read.ahead.block.size | 先読みブロックのサイズです。 | 1048576(1MB) | いいえ |
| fs.cosn.read.ahead.queue.size | 先読みキューの長さです。 | 8 | いいえ |
| fs.cosn.maxRetries | COSへのアクセス中にエラーが発生した場合の最大再試行回数です。 | 200 | いいえ |
| fs.cosn.retry.interval.seconds | 各再試行間の時間間隔です。 | 3 | いいえ |
| fs.cosn.server-side-encryption.algorithm | COS サーバー端末暗号化アルゴリズムを構成し、SSE-C、SSE-COS、SSE-KMSをサポート。デフォルトは空で、暗号化されません。 | なし | いいえ |
| fs.cosn.server-side-encryption.key | • COSのSSE-Cサーバー側暗号化アルゴリズムを有効にする場合、SSE-Cのキーを設定する必要があります。キーの形式はbase64でエンコードされたAES-256キーで、デフォルトは空で暗号化されません。<br>• COSのSSE-KMSサーバー側暗号化アルゴリズムを有効にする場合、この設定項目でKMSのユーザーマスターキーCMKを指定できます。指定しない場合、COSがデフォルトで作成したCMKが使用されます。詳細についてはSSE-KMS暗号化を参照してください。 | なし | いいえ |
| fs.cosn.server-side-encryption.context | COSのSSE-KMSサーバー側暗号化アルゴリズムを有効にする場合、この設定項目で暗号化コンテキストを指定できます。値はJSON形式のキーと値のペアです。 | なし | いいえ |

| | | | |
|---|---|---|---|
| | 例えば： {"key1":"value1"} | | |
| fs.cosn.client-side-encryption.enabled | クライアントサイド暗号化を有効にしますか? デフォルトでは有効になっていません。有効にした場合、クライアントサイド暗号化の公開鍵と秘密鍵を設定する必要があります。この場合、appendおよびtruncateインターフェースは使用できません。 | false | いいえ |
| fs.cosn.client-side-encryption.public.key.path | クライアントサイド暗号化の公開鍵ファイルの絶対パス | なし | いいえ |
| fs.cosn.client-side-encryption.private.key.path | クライアントサイド暗号化の秘密鍵ファイルの絶対パス | なし無 | いいえ |
| fs.cosn.crc64.checksum.enabled | CRC64チェックを有効にするかどうかです。デフォルトでは有効になっていません。現時点では、hadoop fs -checksumコマンドを使用してファイルのCRC64チェックサムを取得することはできません。 | false | いいえ |
| fs.cosn.crc32c.checksum.enabled | CRC32Cチェックを有効にするかどうかです。デフォルトでは有効になっていません。現時点では、hadoop fs -checksumコマンドを使用してファイルのCRC32Cチェックサムを取得することはできません。有効にできるチェック方法は、crc32cかcrc64のうち1つだけです。 | false | いいえ |
| fs.cosn.traffic.limit | アップロード帯域幅の制御オプション。819200 - 838860800 bits/s、デフォルト値は-1です。これは、デフォルトでは制限がないことを意味します。 | なし | いいえ |

### バケット固有の設定項目

背景：異なるリージョンにある各バケットにアクセスする際、バケットごとに設定項目が異なる場合があるため、一度の設定で複数のバケットにアクセス可能にしています。
コア：個別に設定されたバケットレベルの設定項目が優先的に使用されます。個別設定がない場合は元の設定を使用し、元の設定もない場合はデフォルト設定が適用されます。

以下の例は、examplebucket-1250000000に対してfs.cosn.upload.bufferを個別に設定する場合の記述です。

```
fs.cosn.bucket.examplebucket-1250000000.upload.buffer  ***
```

> **説明:**
> ここで、fs.cosn.upload.bufferのサブキー（trim fs.cosn.）はupload.bufferです。したがって、バケット固有の設定項目は `fs.cosn.bucket.<bucket-appid>.<subkey>` となります。

使用方法:

```
hadoop fs -ls cosn://examplebucket-1250000000/
```

**バケット固有設定項目の簡略化（通常バケット向け）**

`cosn://<bucket>/` 形式でアクセスします。
以下の例は、examplebucket-1250000000に対してfs.cosn.upload.bufferを個別に設定する場合の記述です。

```
fs.cosn.userinfo.appid 1250000000
fs.cosn.bucket.examplebucket.upload.buffer  ***
```

> **説明:**
> ここで、fs.cosn.upload.bufferのサブキー（trim fs.cosn.）はupload.bufferです。すでにappidが設定されているため、バケット固有の設定項目は `fs.cosn.bucket.<bucket>.<subkey>` に簡略化できます。

使用方法:

```
hadoop fs -ls cosn://examplebucket/
```

**バケット固有設定項目の簡略化（メタデータアクセラレーションが有効なバケット向け）**

`cosn://<bucket>/` 形式でアクセスします。
以下の例は、examplebucket-1250000000に対してfs.cosn.upload.bufferを個別に設定する場合の記述です。

```
fs.cosn.userinfo.appid 1250000000
fs.cosn.trsf.fs.ofs.use.short.bucketname true
fs.cosn.bucket.examplebucket.upload.buffer  ***
```

> **説明:**

> ここで、fs.cosn.upload.bufferのサブキー（trim fs.cosn.）はupload.bufferです。すでにappidが設定されているため、バケット固有の設定項目は `fs.cosn.bucket.<bucket>.<subkey>` に簡略化できます。

旧バージョンのOFS Java SDKは、マウントポイントの形式として `<bucket-appid>` 形式でのアクセスのみサポートされていたため、最新バージョンの依存プラグインに更新する必要があります。
依存プラグインのバージョン情報：

```
ofs java sdk >= 1.1.8
hadoop cos >= 8.3.0
```

使用方法：

```
hadoop fs -ls cosn://examplebucket/
```

## サーバーサイド暗号化

Hadoop-COSはサーバー側暗号化をサポートしており、現在3つの暗号化方式を提供しています：COSキーを使用したサーバー側の暗号化（SSE-COS）、KMSキーを使用したサーバー側の暗号化（SSE-KMS）、およびお客様が用意したキーでのサーバー側の暗号化（SSE-C）。Hadoop-COSの暗号化機能はデフォルトで無効になっており、お客様が以下の方法で設定を有効にすることができます。

### SSE-COSの暗号化

SSE-COSの暗号化とは、COSホストキーによるサーバー側の暗号化です。Tencent Cloud COSはマスターキーをホストし、データを管理します。ユーザーはHadoop-COSを使用する場合、`$HADOOP_HOME/etc/hadoop/core-site.xml` ファイルに以下の設定を追加すると、SSE-COSの暗号化を実装することができます。

```
<property>
        <name>fs.cosn.server-side-encryption.algorithm</name>
        <value>SSE-COS</value>
        <description>The server side encryption algorithm.</description>
</property>
```

### SSE-KMS

SSE-KMSは、KMSキーを使用したサーバー側の暗号化です。KMSはTencent Cloudが提供するセキュリティ管理サービスで、第三者機関によって認証されたハードウェアセキュリティモジュール（HSM、Hardware Security Module）を使用してキーを生成・保護します。お客様が簡単にキーを作成・管理でき、複数のアプリケーション

やビジネスにおけるキー管理のニーズ、さらには規制・コンプライアンス要件を満たすことができます。Hadoop-COSを使用する際、お客様は `$HADOOP_HOME/etc/hadoop/core-site.xml` ファイルに以下の設定を追加することで、SSE-KMSを実装できます。

```
<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の暗号化

SSE-Cの暗号化とは、ユーザー定義キーによるサーバー側の暗号化です。暗号化鍵はユーザーが提供し、COSはユーザーがオブジェクトをアップロードする際に、ユーザーが提供した暗号化鍵を使用して、ユーザーのデータをAES-256で暗号化します。Hadoop-COSを使用する場合、ユーザーは `$HADOOP_HOME/etc/hadoop/core-site.xml` ファイルに以下の設定を追加して、SSE-Cの暗号化を実装することができます。

```
<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>MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=</value> #ユー
ザーはSSE-Cのキーを自分で設定する必要があり、キーの形式はbase64でエンコードされた
```

```
AES-256キーです。
        <description>The SSE-C server side encryption key.</description>
 </property>
```

**注意：**

- Hadoop-COSのSSE-Cサーバー側の暗号化は、COSのSSE-Cサーバー側の暗号化に依存しています。したがってHadoop-COSは、ユーザーが提供する暗号化鍵を保存しません。また、COSのSSE-Cサーバー側の暗号化方式では、ユーザーが提供した暗号化鍵は保存されず、暗号化鍵にランダムデータが追加されたHMAC値が保存され、この値がオブジェクトにアクセスするためのユーザーのリクエストの認証に使われる点に注意する必要があります。COSは、ランダムデータのHMAC値を使用して暗号化鍵の値を推測したり、暗号化されたオブジェクトの内容を復号したりすることはできません。そのため、ユーザーが暗号化鍵を紛失した場合、このオブジェクトを再取得できなくなります。
- Hadoop-COSがSSE-Cサーバー側暗号化アルゴリズムで設定されている場合、SSE-Cキーはfs.cosn.server-side-encryption.key設定項目で設定しなければなりません。キー形式は、base64でエンコードされたAES-256キーです。
- Hadoop-COS v5.10.0では、書き込み以外のリクエストに対して「Set SSE_COS request no such method」という例外が報告されます。

## クライアントサイド暗号化

COSNクライアントサイド暗号化はRSA暗号化方式を採用しています。キーは公開鍵と秘密鍵に分かれており、公開鍵はファイルの暗号化プロセスに、秘密鍵はファイルの復号プロセスに使用されます。ファイルをアップロードする際、COSNはランダムなキーを生成し、そのキーでファイルを対称暗号化します。さらに公開鍵でこのランダムキーを暗号化し、暗号化された情報をファイルのメタデータに保存します。ファイルをダウンロードする際、COSNは秘密鍵を使用してメタデータから暗号化されたランダムキーを復号し、その復号されたランダムキーでファイルを復号します。公開鍵と秘密鍵はクライアントサイドのローカルでのみ計算処理され、ネットワーク上で転送されたり、サーバーサイドに保存されたりすることはないため、マスターキーのデータセキュリティが保証されます。

- クライアントサイド暗号化機能を利用する際、お客様はマスターキーの完全性と正確性について自ら責任を負うものとします。暗号化データをコピーまたは移行する場合には、暗号化メタデータの完全性と正確性についても、お客様が責任を負うものとします。お客様の不適切な管理により、マスターキーの誤用や紛失、もしくは暗号化メタデータのエラーまたは紛失が発生し、それによって暗号化データが復号不能となり、これに起因する一切の損失および結果については、お客様ご自身で責任を負うものとします。
- クライアントサイド暗号化を有効にすると、appendおよびtruncateインターフェースはサポートされなくなります。
- クライアントサイド暗号化機能が無効になっているクライアントで、暗号化されたファイルに対して `hadoop fs -cp` コマンドを実行すると、暗号化情報が失われます。
- クライアントサイド暗号化を有効にすると、CRCファイル検証と、非同期マルチパートアップロードがデフォ

ルトで無効となります。

Hadoop-COSを使用する際、ユーザーは `$HADOOP_HOME/etc/hadoop/core-site.xml` ファイルに以下の設定を追加することで、SSE-COS暗号化を実装できます。

```
 <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>
```

以下のコードを使用してキーを生成できます。

```
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;

// 非対称キーであるRSAを使用して、毎回生成されるランダムな対称キーを暗号化します。
```

```
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);
    }
}
```

## `yarn-site.xml` のクラスパス設定

`yarn-site.xml` を追加

```
    <property>
       <name>yarn.application.classpath</name>
       <value>$HADOOP_CLASSPATH</value>
    </property>
```

## 使用方法

### ユースケース

コマンド形式は、`hadoop fs -ls -R cosn://<BucketName-APPID>/<パス>` 、または `hadoop fs -ls -R /<パス>` （`fs.defaultFS` オプションは `cosn://BucketName-APPID` に設定する必要あり）です。次の例では、examplebucket-1250000000という名のbucketを例として取り上げ、その後に具体的なパスを追加することができます。

```
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
```

MapReduceに付属しているwordcountを実行し、以下のコマンドを実行します。

> ⚠ 注意：
> 以下のコマンドのhadoop-mapreduce-examples-2.7.2.jarは、バージョン2.7.2を例としています。バージョンが異なる場合は、対応するバージョン番号に変更してください。

```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-
2.7.2.jar wordcount cosn://example/mr/input cosn://example/mr/output3
```

実行に成功すると、以下のような統計情報が返されます。

```
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
```

## JavaコードによるCOSNへのアクセス

```
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();
        // COSNの設定項目については、
https://cloud.tencent.com/document/product/436/6884#hadoop-.E9.85.8D.E7.
BD.AEをご参照ください
        // 以下の設定は、入力必須項目です
        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", "AKXXXXXXXXXXXXXXXXX");
        conf.set("fs.cosn.userinfo.secretKey", "XXXXXXXXXXXXXXXXXX");
        conf.set("fs.ofs.user.appid", "XXXXXXXXXX");
        // その他の設定については、公式サイトのドキュメント
https://cloud.tencent.com/document/product/436/6884#hadoop-.E9.85.8D.E7.
BD.AEをご参照ください
        // CRC64チェックを有効にするかどうかです。デフォルトでは有効になっていませ
ん。この時点では、hadoop fs -checksumコマンドを使用してファイルのCRC64チェックサム
を取得することはできません
        conf.set("fs.cosn.crc64.checksum.enabled", "true");
        String cosnUrl = "cosn://f4mxxxxxxxx-125xxxxxxx";
        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 {
        // ファイルを作成します（存在する場合は上書きします）
        // if the parent dir does not exist, fs will create it!
        FSDataOutputStream out = fs.create(filePath, true);
        try {
            // ファイルに書き込みます
            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());
        }
    }

    // 再帰的削除フラグは、ディレクトリを削除するために用いられます
    // 再帰がfalseで、dirが空でない場合、操作は失敗します
    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 {
        // ファイルの初期化
        FileSystem fs = initFS();

        // ファイルの作成
        Path cosnFilePath = new Path("/folder/exampleobject.txt");
        createFile(fs, cosnFilePath);
```

```
        // ファイルの読み取り
        readFile(fs, cosnFilePath);

        // ファイルまたはディレクトリの照会
        queryFileOrDirStatus(fs, cosnFilePath);

        // ファイルのチェックサムの取得
        getFileCheckSum(fs, cosnFilePath);

        // ローカルからファイルをコピーする
        Path localFilePath = new
Path("file:///home/hadoop/ofs_demo/data/exampleobject.txt");
        copyFileFromLocal(fs, cosnFilePath, localFilePath);

        // ファイルをローカルで取得する
        Path localDownFilePath = new
Path("file:///home/hadoop/ofs_demo/data/exampleobject.txt");
        copyFileToLocal(fs, cosnFilePath, localDownFilePath);

        listDirPath(fs, cosnFilePath);
        // リネーム
        mkdir(fs, new Path("/doc"));
        Path newPath = new Path("/doc/example.txt");
        renamePath(fs, cosnFilePath, newPath);

        // ファイルの削除
        deleteFileOrDir(fs, newPath, false);

        // ディレクトリの作成
        Path dirPath = new Path("/folder");
        mkdir(fs, dirPath);

        // ディレクトリにファイルを作成する
        Path subFilePath = new Path("/folder/exampleobject.txt");
        createFile(fs, subFilePath);

        // ディレクトリのリストアップ
        listDirPath(fs, dirPath);
```

```
        // ディレクトリの削除
        deleteFileOrDir(fs, dirPath, true);
        deleteFileOrDir(fs, new Path("/doc"), true);

        // ファイルシステムを閉じる
        closeFileSystem(fs);
    }
}
```

## よくあるご質問

Hadoopツールの使用に関するご質問は、Hadoopツールに関するよくあるご質問 をご参照ください。

# COSDistCpツール

最終更新日：：2024-06-26 10:27:35

## 機能説明

COSDistCpは、MapReduceをベースとした分散ファイルコピーツールで、主にHDFSとCOS間のデータコピーに使用され、主に次のような機能を備えています。

- 長さ、CRCチェックサムに応じて、ファイルの増分移行とデータチェックを行います
- ソースディレクトリ内のファイルに対して正規表現でのフィルタリングを行います
- ソースディレクトリ内のファイルを解凍し、想定の圧縮形式に変換します
- 正規表現に基づいてテキストファイルを集約します
- ソースファイルとソースディレクトリのユーザー、グループ、拡張属性および時間を保持します
- アラートとPrometheusモニタリングを設定します
- ファイルサイズ分布の統計
- 読み込み帯域幅の速度制限

## 使用環境

### システム環境

Linuxシステムをサポートしています。

### ソフトウェア依存

Hadoop-2.6.0以降、Hadoop-COSプラグイン5.9.3以降。

## ダウンロードとインストール

### COSDistCpjarパッケージの取得

- Hadoop 2.x ユーザーは、cos-distcp-1.12-2.8.5.jarパッケージ をダウンロードし、jarパッケージの MD5 チェックサム に基づき、ダウンロードしたjarパッケージが完全かどうかを確認します。
- Hadoop 3.xユーザーは、cos-distcp-1.12-3.1.0.jarパッケージ をダウンロードし、jarパッケージの MD5 チェックサム に基づき、ダウンロードしたjarパッケージが完全かどうかを確認します。

### インストール説明

Hadoop環境において、Hadoop-COS をインストールすると、直接COSDistCpツールを実行できます。Hadoop-COSプラグインがインストールおよび設定されていない環境のユーザーの場合、Hadoopのバージョンに従って、対応するバージョンのCOSDistCp jar、Hadoop-COS jar、cos_api-bundle jarパッケージをダウンロードし（jarパッケージのダウンロードアドレスは上記参照）、Hadoop-COS関連のパラメータを指定してコ

ピータスクを実行します。jarパッケージアドレスにはローカルjarの存在するアドレスを入力する必要があります。

```
hadoop jar cos-distcp-${version}.jar \
-libjars cos_api-bundle-${version}.jar,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
```

## 原理の説明

COSDistCpはMapReduceフレームワークをベースとして実装されています。これはマルチプロセス+マルチスレッドのアーキテクチャで、ファイルのコピー、データのチェック、圧縮、ファイル属性の保持およびコピーの再試行を行うことができます。COSDistCpは、デフォルトでターゲットの同名の既存ファイルを上書きします。ファイルの移行またはチェックに失敗した場合、対応するファイルのコピーが失敗し、移行に失敗したファイル情報が一時ディレクトリに記録されます。ソースディレクトリにファイルが追加されたり、内容が変更されたりした場合、skipModeやdiffModeモードを使用して、ファイルの長さまたはCRCチェック値を比較することによって、データチェックサムとファイルの増分移行を行うことができます。

## パラメータの説明

hadoopユーザーの下でコマンド `hadoop jar cos-distcp-${version}.jar --help` を使用すると、COSDistCpでサポートされているパラメータオプションを確認することができます。ここで `${version}` はバージョン番号です。以下は現在のバージョンのCOSDistCpのパラメータの説明です。

| プロパティキー | 説明 | デフォルト値 | 入力必須かどうか |
| --- | --- | --- | --- |
| --help | COSDistCpでサポートされているパラメータオプションを出力します<br>例：--help | なし | いいえ |
| --src=LOCATION | コピーのソースディレクトリを指定します。これは | なし | はい |

| | | | |
|---|---|---|---|
| | HDFSまたはCOSパスにすることができます<br>例：--src=hdfs://user/logs/ | | |
| --dest=LOCATION | コピーのターゲットディレクトリを指定します。これは、HDFSまたはCOSパスにすることができます<br>例：--dest=cosn://examplebucket-1250000000/user/logs | なし | はい |
| --srcPattern=PATTERN | 正規表現を指定して、ソースディレクトリにあるファイルをフィルタリングします。<br>例：`--srcPattern='.*\.log$'`<br>**注意：記号 `*` がshellによって解釈されるのを避けるために、パラメータをシングルクォーテーションで囲む必要があります** | なし | いいえ |
| --taskNumber=VALUE | コピープロセス数を指定します。例：--taskNumber=10 | 10 | いいえ |
| --workerNumber=VALUE | コピースレッド数を指定します。COSDistCpは、各コピープロセスでこのパラメータサイズのコピースレッドプールを作成します<br>例：--workerNumber=4 | 4 | いいえ |
| --filesPerMapper=VALUE | Mapper入力ファイル1ファイルあたりの行数を指定します<br>例：--filesPerMapper=10000 | 500000 | いいえ |
| --groupBy=PATTERN | テキストファイルを集約するための正規表現を指定します<br>例：--groupBy='.*group-input/(\d+)-(\d+).*' | なし | いいえ |
| --targetSize=VALUE | ターゲットファイルのサイズをMB単位で指定します。--groupByとともに使用します<br>例：--targetSize=10 | なし | いいえ |
| --outputCodec=VALUE | 出力ファイルの圧縮方法を指定します。gzip、lzo、snappy、none、keepを選択できます。このうち、<br>• keepは、元のファイルの圧縮方法を維持します。<br>noneは、ファイルの拡張子に従ってファイルを解凍します。<br>例：--outputCodec=gzip<br>注意：ファイル /dir/test.gzipと/dir/test.gzがあり、出力形式をlzoに指定する場合は、最終的に1つのファイル/dir/test.lzoだけが維持されます | keep | いいえ |
| --deleteOnSuccess | ソースファイルのターゲットディレクトリへのコピーが成功したら、直ちにソースファイルを削除するよう | false | いいえ |

| | | | |
|---|---|---|---|
| | 指定します<br>例：--deleteOnSuccess,<br>**注意：1.7以降のバージョンではこのパラメータは提供されませんので、データ移行が成功し、--diffModeを使用してチェックを行った後、ソースファイルシステムのデータを削除することをお勧めします** | | |
| --multipartUploadChunkSize=VALUE | Hadoop-COSプラグインがファイルをCOSに転送するときのチャンクサイズを指定します。COSがサポートするチャンクの最大数は10000で、ファイルサイズに応じてチャンクサイズをMB単位で調整できます。デフォルトは8MBです<br>例：--multipartUploadChunkSize=20 | 8MB | いいえ |
| --cosServerSideEncryption | ファイルがCOSにアップロードされる際に、暗号化・復号アルゴリズムとしてSSE-COSを使用するように指定します<br>例：--cosServerSideEncryption | false | いいえ |
| --outputManifest=VALUE | コピーが完了した際、ターゲットディレクトリにそのコピーのターゲットファイルの情報リスト（GZIP圧縮）が発行されるように指定します<br>例：--outputManifest=manifest.gz | なし | いいえ |
| --requirePreviousManifest | 増分をコピーする場合は、--previousManifest=VALUEパラメータを指定する必要があります<br>例：--requirePreviousManifest | false | いいえ |
| --previousManifest=LOCATION | 前回のコピーで生成されたターゲットファイルの情報<br>例：--previousManifest=cosn://examplebucket-1250000000/big-data/manifest.gz | なし | いいえ |
| --copyFromManifest | --previousManifest=LOCATIONとともに使用すると、--previousManifest内のファイルをターゲットファイルシステムにコピーできます<br>例：--copyFromManifest | false | いいえ |
| --storageClass=VALUE | COSタイプを指定します。オプション値は、STANDARD、STANDARD_IA、ARCHIVE、DEEP_ARCHIVE、INTELLIGENT_TIERINGです。サポートされているストレージタイプと概要については、ストレージタイプの概要をご参照ください | なし | いいえ |
| --srcPrefixesFile=LOCATIO | ローカルファイルを指定します。ファイルの各行には、コピーする必要のあるソースディレクトリが含ま | なし | いいえ |

| | | | |
|---|---|---|---|
| N | れます<br>例：--srcPrefixesFile=file:///data/migrate-folders.txt | | |
| --skipMode=MODE | ファイルをコピーする前に、ソースファイルとターゲットファイルが同じかどうかをチェックし、同じであればスキップします。none（チェックしない）、length（長さ）、checksum（CRC値）、length-mtime(長さ+mtime値)、length-checksum（長さ+CRC値）が選択可能です<br>例：--skipMode=length | length-checksum | いいえ |
| --checkMode=MODE | ファイルコピー完了時に、ソースファイルとターゲットファイルが同じかどうかをチェックします。none（チェックしない）、length（長さ）、checksum（CRC値）、length-checksum（長さ+mtime値）、length-checksum（長さ+ CRC値）が選択可能です<br>例：--checkMode=length-checksum | length-checksum | いいえ |
| --diffMode=MODE | ソースディレクトリとターゲットディレクトリの差分ファイルリストを指定して取得します。length（長さ）、checksum（CRC値）、length-checksum（長さ+mtime値）、length-checksum（長さ+CRC値）が選択可能です<br>例：--diffMode=length-checksum | なし | いいえ |
| --diffOutput=LOCATION | diffModeのHDFS出力ディレクトリを指定します。この出力ディレクトリは、必ず空である必要があります<br>例：--diffOutput=/diff-output | なし | いいえ |
| --cosChecksumType=TYPE | Hadoop-COSプラグインが使用するCRCアルゴリズムを指定します。オプション値はCRC32CとCRC64です<br>例：--cosChecksumType=CRC32C | CRC32C | いいえ |
| --preserveStatus=VALUE | ソースファイルのuser、group、permission、xattr、timestampsのメタ情報をターゲットファイルにコピーするかどうかを指定します。オプション値は、ugpxt（user、group、permission、xattr、timestampsの英語頭文字）です<br>例：--preserveStatus=ugpt | なし | いいえ |
| --ignoreSrcMiss | ファイルリストには存在しても、コピー時には存在しないファイルを無視します | false | いいえ |
| --promGatewayAddress=VA | MapReduceタスクによって実行されるCounterデータがプッシュされるPrometheus PushGatewayのア | なし | いいえ |

| | | | |
|---|---|---|---|
| LUE | ドレスとポートを指定します | | |
| --promGatewayDeleteOnFinish=VALUE | タスクが完了したら、Prometheus PushGatewayのJobNameのメトリクスコレクションを削除するよう指定します<br>例：--promGatewayDeleteOnFinish=true | true | いいえ |
| --promGatewayJobName=VALUE | Prometheus PushGatewayに報告するJobNameを指定します<br>例：--promGatewayJobName=cos-distcp-hive-backup | なし | いいえ |
| --promCollectInterval=VALUE | MapReduceタスクのCounter情報を収集する間隔をms単位で指定します<br>例：--promCollectInterval=5000 | 5000 | いいえ |
| --promPort=VALUE | Prometheusメトリクスを外部に公開するサーバーポートを指定します<br>例：--promPort=9028 | なし | いいえ |
| --enableDynamicStrategy | タスクの動的割り当てポリシーを指定して、移行速度の速いタスクがより多くのファイルを移行できるようにします。<br>**注意：このモードには一定の制限があります。例えば、プロセスに異常が発生した場合、タスクカウンターは不正確になります。移行の完了後、--diffModeを使用してデータチェックを行ってください**<br>例：--enableDynamicStrategy | false | いいえ |
| --splitRatio=VALUE | Dynamic Strategyの分割比を指定します。splitRatio値が大きいほど、タスクの粒度は小さくなります<br>例：--splitRatio=8 | 8 | いいえ |
| --localTemp=VALUE | Dynamic Strategyによって生成されたタスク情報ファイルが保存されているローカルフォルダを指定します<br>例：--localTemp=/tmp | /tmp | いいえ |
| --taskFilesCopyThreadNum=VALUE | Dynamic StrategyのタスクファイルをHDFSにコピーする際の同時実行性を指定します<br>例：--taskFilesCopyThreadNum=32 | 32 | いいえ |
| --statsRange=VALUE | 統計の区間範囲を指定します<br>例：---statsRange=0,1mb,10mb,100mb,1gb,10gb,inf | 0,1mb,10mb,100m | いいえ |

| | | | |
|---|---|---|---|
| | | b,1gb,10gb,inf | |
| --printStatsOnly | 移行するファイルサイズの分布情報のみ統計が行われ、データは移行されません<br>例：--printStatsOnly | なし | いいえ |
| --bandWidth | 移行する各ファイルの読み込み帯域幅を制限します。単位はMB/s、デフォルトは-1で、読み込み帯域幅は制限しません。<br>例：--bandWidth=10 | なし | いいえ |
| --jobName | 移行タスクの名前を指定します。<br>例：--jobName=cosdistcp-to-warehouse | なし | いいえ |
| --compareWithCompatibleSuffix | --skipModeと--diffModeパラメータを使用する場合、ソースファイルの拡張子gzipをgzに変換するかどうか、lzopファイルの拡張子をlzoに変換するかどうかを判断します。<br>例：--compareWithCompatibleSuffix | なし | いいえ |
| --delete | ソースディレクトリとターゲットディレクトリのファイルの一致性を保証します。ソースディレクトリにはなく、ターゲットディレクトリにはあるファイルを独立したtrashディレクトリ下に移動させ、同時にファイルリストを生成します。<br>注意：--diffModeパラメータは同時に使用できません | なし | いいえ |
| --deleteOutput | deleteのHDFS出力ディレクトリを指定します。このディレクトリは必ず空でなければなりません<br>例： --deleteOutput=/dele-output | なし | いいえ |

# ユースケース

## helpオプションの確認

パラメータ `--help` によりコマンドを実行し、COSDistCpでサポートされているパラメータを確認します。次に例を示します。

```
hadoop jar cos-distcp-${version}.jar --help
```

上記コマンドにおいて `${version}` はCOSDistCpのバージョン番号です。例えば、バージョン1.0のCOSDistCp jarパッケージ名はcos-distcp-1.0.jarとなります。

## 移行するファイルサイズ分布に関する情報の統計を行います

パラメータ `--printStatsOnly` と `--statsRange=VALUE` によりコマンドを実行すると、移行するファイルのサイズ分布に関する情報が出力されます。

```
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%)             |
```

## 移行するファイルのソースディレクトリとターゲットディレクトリを指定します

パラメータ `--src` と `--dest` によりコマンドを実行します。次に例を示します。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse
```

COSDistCpは、コピーが失敗したファイルをデフォルトで5回再試行します。それでも失敗した場合、失敗したファイル情報を/tmp/${randomUUID}/output/failed/ディレクトリに書き込みます。ここで${randomUUID}は、ランダム文字列です。失敗したファイル情報を記録した後、COSDistcpは残りのファイルの移行を続行しますが、一部のファイルが失敗したために移行タスクが失敗することはありません。移行タスクが完了すると、COSDistcpはカウンター情報（タスクがマシンへ送信され、MapReduceタスクの提出側INFOログの出力が設定されたことを確認してください）を出力し、移行に失敗したファイルが存在するかどうかを判断します。存在した場合は、タスクを送信したクライアントに異常が報告されます。
出力ファイルには、次のようなソースファイル情報が含まれます。

1. ソースファイルのリストに存在しますが、コピー時にソースファイルが存在しないため、SRC_MISSと記録されます

2. その他の原因によるコピーの失敗は、すべてCOPY_FAILEDと記録されます

copyコマンドを再実行して、増分移行を行うことができます。次のコマンドによって、MapReduceタスクのログを取得し、ファイルコピーに失敗した原因を特定します。ここでapplication_1610615435237_0021は、アプリケーションIDです。

```
yarn logs -applicationId application_1610615435237_0021 >
application_1610615435237_0021.log
```

## Countersの確認

コピータスクの終了時に、ファイルのコピーに関する統計情報が出力されます。関連するカウンターは次のとおりです。

```
CosDistCp Counters
        BYTES_EXPECTED=10198247
        BYTES_SKIPPED=10196880
        FILES_COPIED=1
        FILES_EXPECTED=7
        FILES_FAILED=1
        FILES_SKIPPED=5
```

ファイルコピーの統計情報は次のとおりです。

| 統計項目 | 説明 |
| --- | --- |
| BYTES_EXPECTED | ソースディレクトリの統計に基づいてコピーが必要なファイルの合計サイズ。単位：バイト |
| FILES_EXPECTED | ディレクトリファイルを含む、ソースディレクトリの統計に基づいてコピーが必要なファイル数 |
| BYTES_SKIPPED | 長さまたはチェックサム値が等しく、コピーされないファイルサイズの合計。単位：バイト |
| FILES_SKIPPED | 長さまたはチェックサム値が等しく、コピーされないソースファイル数 |
| FILES_COPIED | コピーに成功したソースファイル数 |
| FILES_FAILED | コピーに失敗したソースファイル数 |
| FOLDERS_COPIED | コピーに成功したディレクトリ数 |

| | |
|---|---|
| FOLDERS_SKIPPED | スキップしたディレクトリ数 |

## コピープロセス数と各コピープロセスのコピースレッド数を指定します

パラメータ `--taskNumber` と `--workersNumber` によってコマンドを実行します。COSDistCpはマルチプロセス+マルチスレッドのコピーアーキテクチャを使用しており、次のことが可能です。

- `--taskNumber` でコピーするプロセス数を指定します
- `--workerNumber` で各コピープロセスにおけるコピースレッド数を指定します

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest
cosn://examplebucket-1250000000/data/warehouse --taskNumber=10 --
workerNumber=5
```

## 同じチェックサムを持つファイルをスキップし、増分移行を実行します

パラメータ `--skipMode` によりコマンドを実行します。同じ長さとチェックサムを持つソースとターゲットファイルのコピーをスキップします。デフォルト値はlength-checksumです。

```
hadoop jar cos-distcp-${version}.jar  --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse  --skipMode=length-
checksum
```

`--skipMode` オプションは、ファイルをコピーする前に、ソースファイルとターゲットファイルが同じかどうかをチェックして、同じであればスキップします。none（チェックしない）、length（長さ）、checksum（CRC値）、length-checksum（長さ+CRC値）が選択可能です。
ソースとターゲットのファイルシステムのチェックサムアルゴリズムが異なる場合、ソースファイルを読み込んで新しいチェックサムを計算します。ソースがHDFSの場合、HDFSソースがCOMPOSITE-CRC32Cチェックサムアルゴリズムをサポートしているかどうかは、次のように確認することができます。

```
hadoop fs  -Ddfs.checksum.combine.mode=COMPOSITE_CRC -checksum
/data/test.txt
/data/test.txt  COMPOSITE-CRC32C        6a732798
```

## 移行完了後のデータチェックおよび増分移行

パラメータ `--diffMode` と `--diffOutput` によりコマンドを実行します。

- `--diffMode` のオプション値はlengthとlength-checksumです。
- `--diffMode=length` は、ファイルサイズが同じかどうかによって、差分ファイルのリストを取得するこ

とを表します。

- `--diffMode=length-checksum` 、ファイルサイズとCRCチェックサムが同じかどうかによって、差分ファイルのリストを取得することを表します。
- `--diffOutput` は、diff操作の出力ディレクトリを指定します。
ターゲットファイルシステムがCOSで、ソースファイルシステムのCRCアルゴリズムが異なる場合、COSDistCpはソースファイルをプルしてターゲットファイルシステムのCRCを計算し、同じCRCアルゴリズム値を比較します。以下の例では、移行が完了した後、--diffModeパラメータを使用して、ファイルサイズとCRC値に基づきソースファイルとターゲットファイルが同じであることをチェックしています。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --diffMode=length-
checksum --diffOutput=/tmp/diff-output
```

上記のコマンドの実行が成功すると、ソースファイルシステムのファイルリストを基準にカウンター情報（タスクがマシンへ送信され、MapReduceタスクの提出側INFOログの出力が設定されたことを確認してください）が出力されます。このカウンター情報を基に、ソースとターゲットが同じであるか分析することができます。カウンター情報の説明は、次のとおりです。

1. ソースファイルとターゲットファイルが同じ場合、SUCCESSと記録されます
2. ターゲットファイルが存在しない場合、DEST_MISSと記録されます
3. ソースディレクトリのリストに存在しますが、チェック時にソースファイルが存在しないため、SRC_MISSと記録されます
4. ソースファイルとターゲットファイルのサイズが異なる場合、LENGTH_DIFFと記録されます
5. ソースファイルとターゲットファイルのCRCアルゴリズムの値が異なる場合、CHECKSUM_DIFFと記録されます
6. 読み込み権限が不十分であるなどの要因により、diff操作が失敗した場合、DIFF_FAILEDと記録されます
7. ソースがディレクトリ、ターゲットがファイルの場合、TYPE_DIFFと記録されます

また、COSDistcpは、HDFSの `/tmp/diff-output/failed` ディレクトリ（1.0.5および以前のバージョンでは/tmp/diff-output）に差分ファイルリストを発行します。次のコマンドを使用して、SRC_MISS以外の差分ファイルリストを取得することができます。

```
hadoop fs -getmerge /tmp/diff-output/failed diff-manifest
grep -v '"comment":"SRC_MISS"' diff-manifest |gzip > diff-manifest.gz
```

次のコマンドを実行し、差分ファイルリストに基づいて増分移行を実行します。

```
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
```

増分移行が完了したら、再度--diffModeパラメータ付きのコマンドを実行し、ファイルが同じであることをチェックします。

## ソースファイルとターゲットファイルのCRCが同じであるかチェックします

パラメータ `--checkMode` によりコマンドを実行します。ファイルのコピーが完了したときに、ソースファイルとターゲットファイルの長さとチェックサムが同じであることをチェックします。デフォルト値はlength-checksumです。
非COSファイルシステムからCOSに同期する場合やソースのCRCアルゴリズムがHadoop-COSのCRCアルゴリズムと一致しない場合、コピー中にCRCが計算され、コピー完了時にターゲットCOSファイルのCRCを取得し、算出されたソースファイルのCRCと照合します。

```
hadoop jar cos-distcp-${version}.jar   --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --checkMode=length-
checksum
```

> ⚠ 注意：
> --groupByが指定されておらず、outputCodecがデフォルト値の場合に有効になります。

## 単一ファイルの読み込み帯域幅を制限します

パラメータ `--bandWidth` によりコマンドを実行します。単位はMBです。移行した各ファイルの読み込み帯域幅を10MB/sに制限します。次に例を示します。

```
hadoop jar cos-distcp-${version}.jar  --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --bandWidth=10
```

## 複数ディレクトリの同期

ローカルファイル（srcPrefixes.txtなど）を新規作成し、このファイルに移行が必要なディレクトリの絶対パスを追加します。これらのディレクトリ間に親子関係は必要なく、追加すれば、次の例に示すように、catコマンドで確認することができます。

```
cat srcPrefixes.txt
/data/warehouse/20181121/
/data/warehouse/20181122/
```

`--srcPrefixesFile` パラメータを使用してこのファイルを指定し、移行コマンドを実行します。

```
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
```

- 入力ファイルの正規表現でのフィルタリング

パラメータ `--srcPattern` でコマンドを実行すると、 `/data/warehouse/` ディレクトリにある.logで終わるログファイルのみを同期します。次に例を示します。

```
hadoop jar cos-distcp-${version}.jar  --src /data/warehouse/ --dest
cosn://examplebucket-1250000000/data/warehouse --srcPattern='.*\.log$'
```

.tempまたは.tmpで終わるファイルは移行されません。

```
 hadoop jar cos-distcp-${version}.jar --src /data/warehouse/ --dest
cosn://examplebucket-1250000000/data/warehouse/ --srcPattern='.*(?
<!\.temp|\.tmp)$'
```

## Hadoop-COSのファイルチェックとタイプの指定

パラメータ `--cosChecksumType` によりコマンドを実行します。デフォルトはCRC32Cで、オプションはCRC32CとCRC64です。

```
hadoop jar cos-distcp-${version}.jar  --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --cosChecksumType=CRC32C
```

## COSファイルのストレージタイプの指定

パラメータ `--storageClass` によりコマンドを実行します。次に例を示します。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --
outputManifest=manifest-2020-01-10.gz --storageClass=STANDARD_IA
```

## ターゲットファイルの圧縮タイプの指定

パラメータ `--outputCodec` によりコマンドを実行します。このパラメータを使用すると、HDFS内のデータをリアルタイムに圧縮してCOSにバックアップし、ストレージコストを節約することができます。パラメータの

オプション値は、keep、none、gzip、lzop、snappyです。noneオプションはターゲットファイルを非圧縮状態で保存し、keepはオリジナルファイルの圧縮状態を維持します。次に例を示します。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse/logs --dest
cosn://examplebucket-1250000000/data/warehouse/logs-gzip --
outputCodec=gzip
```

> ⚠ 注意：
> keepオプションを除いて、まずファイルを解凍し、それからターゲットの圧縮タイプに変換します。そのためkeepオプション以外では、圧縮パラメータなどの不整合により、ターゲットファイルがソースファイルと一致しないことがありますが、解凍後のファイルは同じになります。--groupByが指定されておらず、--outputCodecがデフォルトの場合は、--skipModeで増分の移行を行い、--checkModeでデータチェックを行うことができます。

## ソースファイルの削除

パラメータ `--deleteOnSuccess` によりコマンドを実行し、 `/data/warehouse` ディレクトリ内のファイルをHDFSからCOSに同期してから、直ちにソースディレクトリ内の対応するファイルを削除します。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --deleteOnSuccess
```

> ⚠ 注意：
> このオプションを指定すると、ファイルが移行されるたびに対応するソースファイルが直ちに削除されます。全体の移行が完了した後、ソースファイルを削除する必要はありませんので、ご注意ください。1.7以降のバージョンでは、このパラメータは提供されません。

## ターゲットリストファイルの発行と最終リスト出力ファイルの指定

パラメータ `--outputManifest` と `--previousManifest` によりコマンドを実行します。

- `--outputManifest` このオプションは、まずローカルでgzip圧縮されたmanifest.gzを発行し、移行が成功したときに `--dest` で指定されたディレクトリに移動させます。
- `--previousManifest` は前回の `--outputManifest` 出力ファイルを指定します。COSDistCpは、同じサイズのファイルをスキップします。

```
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
```

> ⚠ 注意:
> 上記のコマンドの増分移行では、サイズが変更されたファイルのみを同期できますが、ファイル内容が変更されたファイルは同期できません。ファイルの内容が変更される可能性がある場合は、 --diffMode の使用例を参照して、ファイルのCRCに基づいて変更されたファイルのリストを決定してください。

## 移行タスクの割り当てポリシーを動的割り当てに指定

ごく少量の非常に大きなファイル、多数の小さなファイル、移行用マシンの負荷が変化する場合などのように、ファイルサイズが極端に分かれている場合は、enableDynamicStrategy`を使用して、タスクの動的割り当てポリシーを有効にすると、実行速度の速いタスクがより多くのファイルを移行できるようになり、タスクの実行時間が短縮されます。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse    --dest
cosn://examplebucket-1250000000/data/warehouse --enableDynamicStrategy
```

移行完了後の移行データのチェックを行います。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --diffMode=length-
checksum --diffOutput=/tmp/diff-output
```

> ⚠ 注意:
> このモードには一定の制限があります。例えば、プロセス例外が発生した場合、タスクカウンターはカウントが不正確になる可能性があります。移行完了後に--diffModeを使用してデータをチェックしてください。

## ファイルのメタ情報のコピー

パラメータ `--preserveStatus` によりコマンドを実行し、ソースファイルまたはソースディレクトリのuser、group、permission、timestamps（modification timeとaccess time）をターゲットファイルまたはターゲットディレクトリにコピーします。このパラメータは、HDFSからCHDFSへファイルをコピーするときに有効になります。
例:

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --preserveStatus=ugpt
```

## Prometheusモニタリングの設定

Yarnの管理画面では、すでに移行が完了したファイル数、バイト数などの情報を含む、COSDistcp移行タスクのカウンターを確認することができます。移行タスクカウンターの曲線の推移をより見やすくするためには、prometheus.ymlを設定してクロールタスクを追加するといった簡単な設定だけで、Prometheus + Grafanaモニタリングシステムで COSDistcp移行タスクのカウンターを表示させることができます。

```
- job_name: 'cos-distcp-hive-backup'
    static_configs:
      - targets: ['172.16.16.139:9028']
```

パラメータ `--promPort=VALUE` によりコマンドを実行し、現在のMapReduceタスクのカウンターを外部に公開します

```
hadoop jar cos-distcp-${version}.jar  --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --promPort=9028
```

事例の Grafana Dashboard をダウンロードしてインポートすると、Grafanaは次のように表示されます。


## ファイルコピー失敗時のアラート

パラメータ `--completionCallbackClass` でコールバッククラスパスを指定してコマンドを実行します。COSDistCpはコピータスクが完了すると、収集したタスク情報をパラメータとしてコールバック関数を実行しま

す。ユーザー定義のコールバック関数については、以下のインターフェースを実装する必要がありますので、コールバックのサンプルコードのダウンロード に移動してください。

```
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は、内部でBCMのアラートを統合し、タスクに異常が発生したり、ファイルコピーに失敗したりした場合にアラートを実行します。

```
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.qcloud.cos.distcp.DefaultTaskCompletionCallb
ack
```

上記コマンドのalarmPolicyIdは、BCMアラートポリシーであり、BCMコンソールで作成と設定ができます（アラート管理 > アラート設定 > カスタムメッセージ）。

## よくあるご質問

### COSDistcpを使用したHDFSデータの移行にはどのような段階があり、どのように移行パフォーマンスを調整してデータの正確性を確保すればよいですか。

COSDistcpでは各ファイルの移行完了ごとに、checkModeに基づいて、移行したファイルに対してチェックが行われます。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse --taskNumber=20
```

また、移行が完了したら、以下のコマンドを実行し、ソースとターゲットの差分ファイルリストを確認することができます。

```
hadoop jar cos-distcp-${version}.jar --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse/ --diffMode=length-
checksum --diffOutput=/tmp/diff-output
```

### Hadoop-COSが設定されていない環境で、COSDistCpを実行するにはどうすればいいですか。

Hadoop-COSプラグインが設定されていない環境のユーザーの場合、Hadoopのバージョンに従って、対応するバージョンのCOSDistCp jarパッケージをダウンロードし、Hadoop-COS関連のパラメータを指定してコピータスクを実行します。

```
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
```

## コピー結果に、一部のファイルのコピーに失敗したと表示されます。どうすればよいですか。

COSDistCpは、ファイルコピー中に発生した IOExceptionに対して5回再試行します。5回目のコピーも失敗した場合は、失敗したファイル情報を `/tmp/${randomUUID}/output/failed/` ディレクトリに書き込みます。ここで、${randomUUID}はランダムな文字列です。コピーが失敗する一般的な原因としては、次のようなものがあります。

1. コピーリストにソースファイルは存在しますが、コピー時にソースファイルが存在しないため、SRC_MISSと記録されます。
2. タスクを開始したユーザーが、ソースファイルの読み込みやターゲットファイルへの書き込みの権限を持っていないことなどが原因で、COPY_FAILEDと記録されます。

ログ情報にソースファイルが存在しないことが記録されており、ソースファイルを実際に無視できる場合は、次のコマンドを使用して、SRC_MISS以外の差分ファイルのリストを取得することができます。

```
hadoop fs -getmerge /tmp/${randomUUID}/output/failed/ failed-manifest
grep -v '"comment":"SRC_MISS"' failed-manifest |gzip > failed-
manifest.gz
```

SRC_MISS以外に失敗したファイルがある場合は、 `/tmp/${randomUUID}/output/logs/` ディレクトリにまとめられた例外ログ情報と、yarnアプリケーションなどのプルアプリケーションログに基づいて原因を診断することができます。次のコマンドを使用できます。

```
yarn logs -applicationId application_1610615435237_0021 >
application_1610615435237_0021.log
```

ここでapplication_1610615435237_0021は、アプリケーションIDです。

## COSDistCpは、ネットワークの異常などが発生した場合、不完全なファイルのコピーは生成されますか。

COSDistCpは、ネットワークの異常、ソースファイルの欠落、権限不足などの場合、ターゲット側とソース側で同じサイズのファイルを発行することはできません。

- COSDistCpのバージョンが1.5以下の場合、COSDistCpはターゲット側で生成されたファイルの削除を試みます。削除に失敗した場合は、コピータスクを再実行してこれらのファイルを上書きするか、不完全なファイルを手動で削除する必要があります。
- COSDistCp 1.5以降のバージョンで、実行環境のHadoop COSプラグインバージョンが5.9.3以降の場合、COSDistCpはCOSへのコピーが失敗すると、abortインターフェースを呼び出してアップロード中のリクエストを終了させるようにします。そのため異常が発生しても、不完全なファイルは発行されません。
- COSDistCp 1.5以降のバージョンで、実行環境のHadoop COSプラグインバージョンが5.9.3未満の場合は、5.9.3以降のバージョンにアップグレードすることをお勧めします。
- COS以外のターゲット側では、COSDistCpはターゲットファイルの削除を試みます。

## COSバケットに、目に見えず完了していないアップロードファイルがあり、ストレージ容量を占有しています。どうすればよいですか。

マシンの異常やプロセスのKillなどの要因により、COSバケット内でフラグメントファイルがストレージ容量を占有する場合があります。この場合、公式ウェブサイトの ライフサイクルドキュメント を参照して、フラグメントの削除ルールを設定し、クリーンアップを行うことができます。

## 移行プロセスでは、メモリオーバーフローとタスクタイムアウトが発生した場合、どのようにパラメータチューニングを行いますか。

移行プロセスでは、COSDistcpとCOSへのアクセスとCHDFSのツールは、 自身のロジックに基づき、メモリの一部を占有します。メモリオーバーフローとタスクタイムアウトを防ぐため、下の例のようにしてMapReduceタスクのパラメータ調整をすることができます。

```
hadoop jar cos-distcp-${version}.jar -Dmapreduce.task.timeout=18000 -
Dmapreduce.reduce.memory.mb=8192 --src /data/warehouse --dest
cosn://examplebucket-1250000000/data/warehouse
```

そのうち、タスクのタイムアウト時間mapreduce.task.timeoutを18000秒に調整することで、大容量のファイルをコピーするとき、タスクタイムアウトの発生を防ぐことができます。Reduceプロセスのメモリ容量mapreduce.reduce.memory.mbサイズを8GBに調整することで、メモリオーバーフローを防ぐことができます。

## 専用回線の移行で、どのように移行タスクの移行帯域幅をコントロールしますか。

COSDistcpでは移行の総帯域幅制限計算式は、taskNumber * workerNumber * bandWidthです。workerNumberを1に設定し、パラメータtaskNumberによる移行の同時実行数のコントロール、およびパラメータbandWidthに

よる単一の同時実行帯域幅のコントロールができます。

# HDFS TO COSツール

最終更新日：：2024-06-26 10:27:35

## 機能説明

HDFS TO COSツールは、HDFSからTencent Cloud COSにデータをコピーするときに使います。|

## 使用環境

### システム環境

LinuxまたはWindowsシステム。

### ソフトウェア依存

JDK 1.7または1.8。

### インストールと設定

環境のインストールと設定の詳細については、Javaのインストールと設定をご参照ください。

## 設定方法

1. Hadoop-2.7.2以降のバージョンをインストールします。具体的なインストール手順については、Hadoopのインストールとテストをご参照ください。
2. GitHubからHDFS TO COSツールをダウンロードし、解凍してください。
3. 同期するHDFSクラスターのcore-site.xmlをconfフォルダにコピーします。このうちcore-site.xmlには、NameNodeの設定情報が含まれています。
4. 設定ファイルcos_info.confを編集して、バケット(Bucket)、リージョン(Region)およびAPIキー情報を保存します。このうちバケット名は、ユーザー定義の文字列と、システムが発行したAPPID文字列をハイフンで連結することで構成されます（例：examplebucket-1250000000）。
5. コマンドラインパラメータで設定ファイルの場所を指定します。デフォルトの場所はconf/cos_info.confです。

> **注意：**
> コマンドラインパラメータのパラメータが設定ファイルと重複している場合は、コマンドラインが優先されます。

## 利用方法

> **説明：**
> 以下は、Linuxを例とした使用方法です。

### ヘルプの確認

```
./hdfs_to_cos_cmd -h
```

### ファイルのコピー

- HDFSからCOSにコピーします。COSにすでに同名のファイルが存在する場合、元のファイルは上書きされます。

```
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/
```

- HDFSからCOSにコピーします。COSにすでに同名で同じ長さのファイルが存在する場合、アップロードは無視されます（1回目のコピー後、再度コピーする場合に適用されます）。

```
./hdfs_to_cos_cmd --hdfs_path=/tmp/hive --cos_path=/hdfs/20170224/ -
skip_if_len_match
```

Hadoop上でファイルサマリーを計算するとオーバーヘッドが大きくなるため、ここでは長さのみを判断しています。

- HDFSからCOSにコピーします。HDFSにHarディレクトリ（Hadoop Archiveアーカイブファイル）が存在する場合、--decompress_harパラメータを指定することでharファイルを自動的に解凍できます。

```
./hdfs_to_cos_cmd --decompress_har --hdfs_path=/tmp/hive --
cos_path=/hdfs/20170224/
```

--decompress_harパラメータを指定しない場合、デフォルトで通常のHDFSディレクトリがコピーされます。すなわち、.harディレクトリ内のindexやmasterindexなどのファイルがそのままコピーされるということです。

### ディレクトリ情報

```
conf : 設定ファイル。core-site.xmlとcos_info.confを保存するときに使います
log  : ログディレクトリ
src  : Javaソースプログラム
dep  : 発行した実行可能なJARパッケージをコンパイルします
```

## 質問とヘルプ

### 設定情報について

バケット(Bucket)、リージョン(Region)、APIキー情報など、入力された設定情報が正しいことを確認してください。このうちバケット名は、ユーザー定義の文字列と、システムが発行したAPPID文字列をハイフンが連結することで構成されます（例：examplebucket–1250000000）。また、マシンの時刻が北京の時刻と一致していることを確認してください（1分程度の差は正常です）。差が大きい場合は、マシンの時刻をリセットしてください。

### DataNodeについて

DataNodeについては、コピープログラムが配置されているマシンも接続できることを確認してください。NameNodeには接続するパブリックネットワークIPがありますが、取得したblockが配置されているDateNodeマシンはプライベートネットワークIPであり、直接接続することはできません。したがって、NameNodeとDataNodeの両方にアクセスできるように、Hadoopのいずれかのノードで同期プログラムを実行することをお勧めします。

### 権限について

Hadoopコマンドを使用してファイルをダウンロードし、正常かどうかチェックしてから、同期ツールを使用してHadoopのデータサポートを同期してください。

### ファイルの上書きについて

COSにすでに存在するファイルについては、デフォルトで再送信と上書きが行われます。ユーザーが明示的に–skip_if_len_matchを指定しない限り、ファイルの長さが同じである場合、アップロードはスキップされます。

### cos pathについて

cos pathのデフォルトはディレクトリであり、最終的にHDFSからコピーされるファイルはこのディレクトリに保存されます。

### Tencent Cloud EMR HDFSからのデータのコピーについて

Tencent Cloud EMR HDFSからCOSにデータをコピーする場合、高性能のDistcpツールを使用することをお勧めします。HadoopファイルシステムとCOS間のデータ移行 をご参照ください。

# オンラインツール (Onrain Tsūru)
# COSリクエストツール

最終更新日：：2024-06-26 10:27:35

## 機能の説明

COSリクエストツールはTencent Cloud Object Storage（COS）が提供するWeb端末デバッグツールです。Tencent Cloud API 3.0 Explorerプラットフォームに統合され、APIインターフェースのデバッグ作業に使用できます。

> ⚠ 注意：
> COSリクエストツールが送信したリクエストは実際にCOSの業務サーバーに送信されます。**すべての操作は実際の操作に等しいため、DELETE系の操作を選択する場合は慎重に操作してください。**

現在COSリクエストツールがサポートしているAPIバージョンはXML APIです。JSON APIバージョンはサポートしていません。

- JSON APIはTencent Cloud COSサービスがXML APIを打ち出す前にユーザーにCOSにアクセスして使用するために提供したAPIインターフェースです。インターフェースと標準XMLのAPIの基礎となるアーキテクチャは同じで、データは相互接続し、互いに利用することができますが、インターフェースに互換性はありません。
- XML APIにはより豊富な機能と特徴があります。Tencent Cloud COSはXML APIバージョンにアップグレードすることを強くお勧めします。

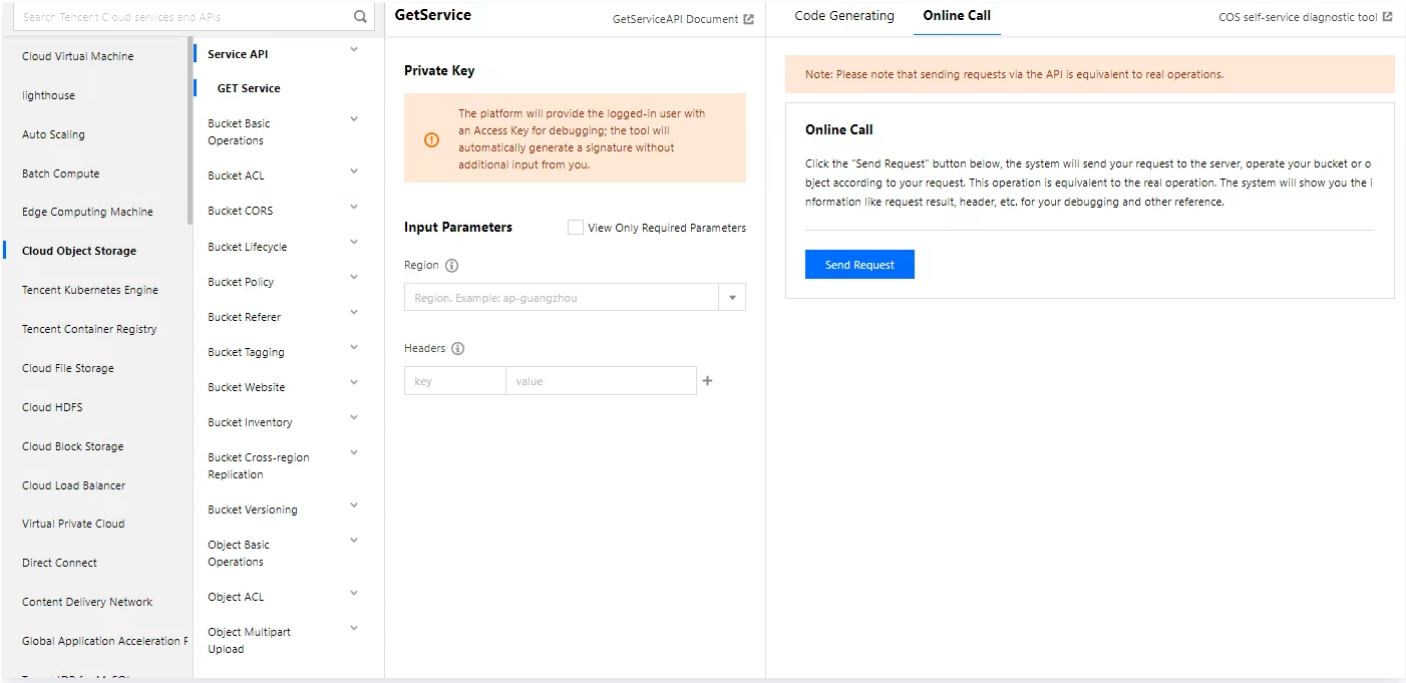
## ツールアドレス

COSリクエストツールをクリックして進みます。

## 使用方法

COS製品を選択し、必要なAPIインターフェースを選択し、そのインターフェース下で対応するパラメータを入力し、リクエストの送信をクリックして対応するリクエストのレスポンス結果を取得します。
COSリクエストツールのページ全体は、左から右へ順に製品欄，インターフェース欄，パラメータ欄および結果欄です。下図のようにそれぞれのカテゴリーで対応する操作を実行でき、最後に結果欄でリクエストを送信して

レスポンス結果および関連するプロセスパラメータ情報を取得できます。




COSリクエストツールに関する詳細な操作は以下の手順をご確認ください。

**1. COS製品の選択**

一番左側の製品欄内で**COS**をクリックすると、APIインターフェース欄内のCOSに関連するAPIインターフェースを確認することができます。

> **説明:**
> COSリクエストツールはTencent Cloud API3.0プラットフォームに統合され、このプラットフォームにはその他のTencent Cloud製品のAPI デバッグツールが搭載されています。同じようにニーズに応じてこのプラットフォーム上でその他の製品を選択し、対応するインターフェースをデバッグすることができます。

**2. デバッグする必要があるAPIインターフェースを選択します**

ニーズに応じて対応するAPIインターフェースを選択してデバッグを行うことができます。APIインターフェース欄には3つのクラスとCOSに関連するAPIインターフェースが表示されます： Serviceインターフェース、Bucketインターフェースおよびobjectインターフェース。

- Serviceクラスのインターフェースは、例えばGET Serviceです。このAPIアカウント下の全てのバケット情報を一覧表示し、APIキー情報を入力する必要があります。アカウントの指定領域内のバケット情報を取得する必要がある場合は、パラメータ欄内で対応するリージョンを選択できます。このAPIの詳細情報について知りたい場合は、 GET Service ドキュメントをご参照ください。
- Bucketクラスのインターフェースには、バケットに操作を行うAPIインターフェース、例えばPUT Bucket lifecycleが含まれます。より詳しくBucketクラスのインターフェースについて知りたい場合は、Bucketインターフェースをご参照ください。
  Objectクラスのインターフェースには、各オブジェクトに操作を行うことができるAPIインターフェース、例

えばPUT Objectが含まれます。より詳しくObjectクラスのインターフェースについて知りたい場合は、Objectインターフェースをご参照ください。

**3. APIに必要なパラメータ情報の入力**

パラメータ欄には、選択したAPIインターフェースの入力必須パラメータが表示されます。COSに関する各APIインターフェースのパラメータ情報については、 APIドキュメント を参照し、ご確認ください。

APIキー情報はAPIインターフェースを呼び出す段階で入力必須のパラメータです。APIインターフェースを使用してバケットまたはオブジェクトなどのリソースを操作する際は、APIキー情報を入力し、今回のAPI リクエスト操作を許可する必要があります。CAMコンソールの APIキー管理 ページにアクセスしてAPIキー情報を取得することができます。

> **説明:**
> 各APIインターフェースに対して、COSリクエストツールは各入力必須のパラメータ項目の後にすべて赤い星マークを追加してそのパラメータが入力必須項目であることを通知します。**入力必須パラメータのみを表示**にチェックを入れて、パラメータ欄の入力必須ではないパラメータの表示を減らすこともできます。

**4. リクエストの送信およびレスポンス結果の確認**

APIを選択し、対応するパラメータを入力した後、**オンラインコール**のオプションタブで**リクエストの送信**をクリックします。この時システムはリクエストをサーバーに送信し、リクエストに応じてバケットまたはオブジェクトリソースを操作します。

> **注意:**
> COSリクエストツールが送信したリクエストを実際にCOSの業務サーバーに送信します。**すべての操作は実際の操作に等しいため、DELETE系の操作を選択する場合は慎重に操作してください。**

リクエストの送信後に、返された結果およびリクエストパラメータは結果欄の下部に表示されます。結果欄下部では、**リクエストパラメータ**でHTTPリクエストボディ情報を確認できます。**レスポンス結果**で今回のリクエストのレスポンスボディ情報が確認できます。**署名プロセス**で今回のリクエストに関する署名およびその生成プロセスが確認でき、同時にシステムは**Curl**でCurl呼び出しのステートメントを提供します。

**事例**

GET Objectを例に取ると、リクエストの送信によって名前が0001.txtのファイルを取得したい場合、**リクエストパラメータ**内に今回のリクエストのリクエストパラメータ情報が表示されます。今回のリクエストのリクエスト例は次のとおりです。

```
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****
```

最初の行に表示されるのはHTTP Verbおよびアクセスのリンクです。次の行に表示されるのはアクセスするドメイン名です。最後の行に表示されるのは今回のリクエストの署名情報です。PUTクラスのリクエストは、そのリクエストヘッダー情報が複雑ですが、同様にいくつかのパブリックリクエストヘッダーが存在します。パブリックリクエストヘッダーに関する情報は、 パブリックリクエストヘッダー をご参照ください。
**署名プロセス**では今回のリクエストに関する署名およびその生成プロセスを確認することができます。署名アルゴリズムに関する詳細の紹介はリクエスト署名 を参照し、詳細をご確認ください。リクエスト署名に生成およびデバッグを行う場合は、COS署名ツールを使用することをお勧めします。
COSが返したレスポンス結果は以下のとおりです。

```
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!
```

最初の行の200 OKはそのリクエストが返された状態コード情報を表し、リクエストが失敗した場合は、対応するエラーコードを返します。エラーコードに関する詳細情報は、 エラーコード ドキュメントをご参照ください。
その後はレスポンスヘッダー情報で、異なるAPIのレスポンスボディにはいずれも違いがありますが、いくつかのパブリックレスポンスヘッダー情報が存在します。パブリックレスポンスヘッダーに関する詳細情報は、 パブリックレスポンスヘッダー をご参照ください。

## 注意事項

**リクエストの送信**をクリックして、入力必須のパラメータを入力したリクエストがCOSサーバーに送信されたことを確認すると、COSはバケットとオブジェクトに対応する操作を行います。この操作は取り消し、ロールバックする事ができません。慎重にご検討ください。

# セルフ診断ツール

最終更新日：：2024-06-26 10:27:35

## 機能の説明

Cloud Object Storage(COS)自己診断ツールは、Tencent Cloud COSがユーザー向けに提供するWebツールで、エラーリクエストのセルフチェックとトラブルシューティングができます。ツール画面でリクエストのRequestIdを入力し（取得方法はRequestId取得の操作ガイドをご参照ください）、診断をクリックすると、ツールはリクエストのインテリジェント診断を行います。また、このリクエストの基本情報やヘルプガイドおよび診断プロンプトを表示して、COSAPIを使用した際のエラーをすばやく見つけるのに役立ちます。

## ツールアドレス

ここをクリックしてCOS自己診断ツールに進みます。

## 利用方法

1. COS自己診断ツールをクリックして、「COS自己診断ツール」画面に進みます。
2. 上部のRequestId入力ボックスに診断するRequestIdを入力し、【診断の開始】をクリックします。
3. しばらくすると、対応するインテリジェント診断の結果が表示されます。

診断結果には、「診断結果」と「リクエスト情報」という2つの部分があります。

- 診断結果は、診断後のアドバイスやヘルプをリクエストするためのものです。アドバイスとヘルプに従って、COS APIのエラーをすばやく特定することができます。
- リクエスト情報は、この RequestIdに対応する基本情報です。

4. 診断結果をフィードバックします。
   診断結果の下にある【役に立った】または【役に立たなかった】をクリックして、診断結果に関するフィードバックをお送りください。いただいたご意見は、ツールの最適化を引き続き行うために役立ててまいります。

## よくあるご質問

このツールは、RequestIdを使用した診断トラブルシューティングに加え、COS APIの使用と、それに関連する問題をチェックするために、よくあるご質問のサンプルと対応するソリューションも豊富にご提供しています。また、ご不明な点がございましたら、お問い合わせをご利用ください。

## 注意事項

標準的なCOSリクエストRequestIdには、次のような特徴があります。

1. 大文字のNから始まる。
2. 長さは30文字以上でなければならない。
3. base64の文字規格に準拠している。