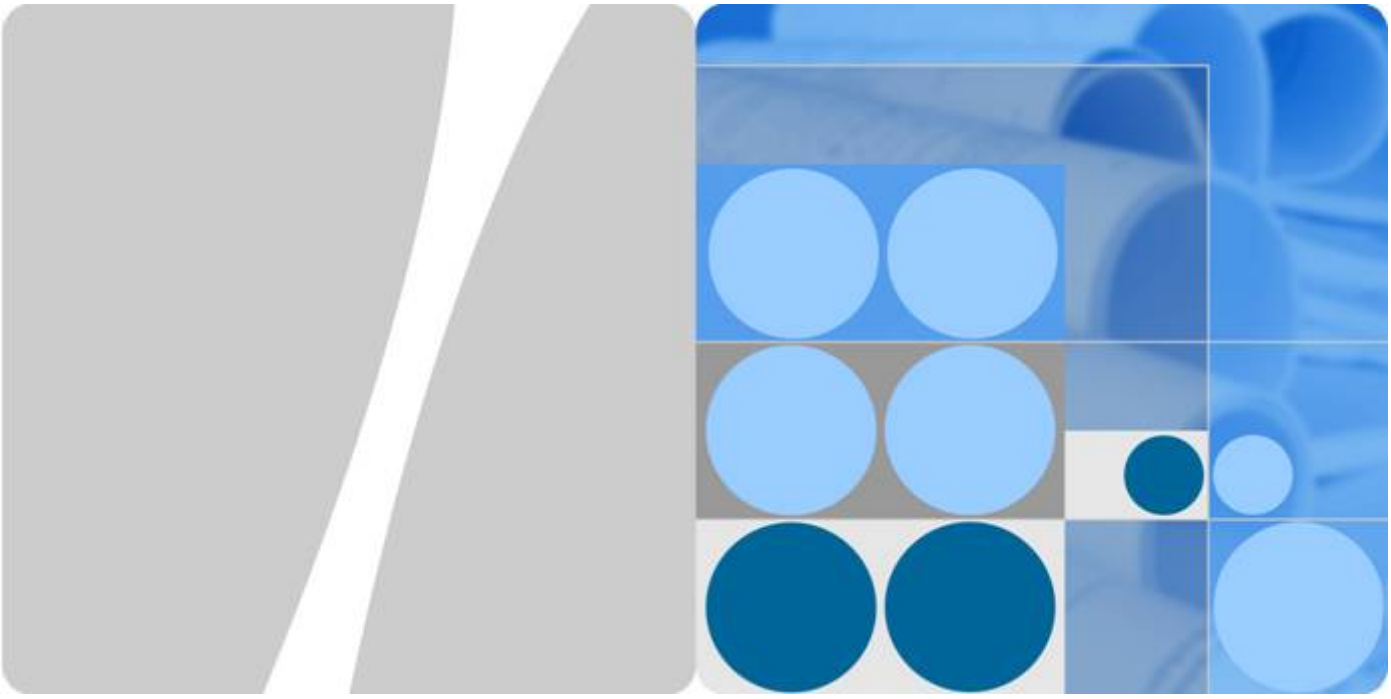


Object Storage Service

Tools Guide (obsutil)

Date **2019-09-30**

Contents

1 Introduction to obsutil

obsutil is a command line tool for accessing and managing OBS. You can use this tool to perform common configurations in OBS, such as creating buckets, uploading and downloading files/folders, and deleting files/folders. If you are familiar with command line interface (CLI), obsutil is recommended as an optimal tool for batch processing and automated tasks.

obsutil is compatible with the Windows, Linux, and macOS operating systems (OSs). **Table 1-1** lists the recommended OS versions.

Table 1-1 Recommended OS versions for using obsutil

OS	Recommended Version
Windows	● Windows 7 ● Windows 8 ● Windows 10 ● Windows Server 2016
Linux	● SUSE 11 ● EulerOS 2
macOS	macOS 10.13.4

Open-Source Addresses

- Click **here** to download the source code of the latest version.
- Click **here** to download the compiled software packages of historical versions.

Tool Advantages

obsutil features the following aspects:

1. Simple and easy to use
2. Lightweight and installation-free
3. Compatible with Windows, Linux, and macOS operating systems

4. Diversified configurations and excellent performance

Application Scenarios

- Automated backup and archiving, for example, periodically uploading local data to OBS.
- Scenarios that cannot be implemented using other tools such as OBS Browser, for example, synchronously uploading, downloading, and copying objects.

Functions

Table 1-2 lists obsutil functions.

Table 1-2 obsutil functions

Function	Description
Basic operations on buckets	Create buckets of different storage classes in specific regions, delete buckets, and obtain the bucket list and configuration information.
Basic operations on objects	Manage objects, including uploading, downloading, deleting, and listing objects. Supported operations are detailed as follows: ● Upload one or more files or folders. ● Upload large files in multiple parts. ● Synchronously upload, download, and copy incremental objects. ● Copy a single object or copy multiple objects in batches by object name prefix. ● Move a single object or move objects in batches by object name prefix. ● Resume failed upload, download, or copy tasks.
Logging	Allows you to configure logging on the client side to record operations on buckets and objects for statistics analysis later.

Command Line Structure

The obsutil command line structures are as follows:

- In Windows

  obsutil *command [parameters...] [options...]*
- In Linux or macOS

  ./obsutil *command [parameters...] [options...]*

NOTE

- **command** indicates the command to be executed, for example, **ls** or **cp**.
- **parameters** indicates the basic parameters (mandatory) of the command, for example, bucket name when creating a bucket.
- **options** indicates the additional parameters (optional) of the command. Additional parameters must be preceded with a hyphen (-) when you run the command.
- The square brackets ([]) are not part of the command. Do not enclose parameter values with them when entering a command.
- If the command contains special characters including ampersands (&), angle brackets (<) and (>), and spaces, they need to be escaped using quotation marks. Use single quotation marks for Linux or macOS and quotation marks for Windows.
- Additional parameters can be input in the **-***key*=*value* or **-***key value* format, for example, **-acl=private**, or **-acl private**. There is no difference between the two formats. Select either one as you like.
- In Windows, you can directly execute **obsutil.exe** to enter an interactive command mode. In this mode, you can input *command [parameters...] [options...]* without **obsutil** to run a command. An example is provided as follows:

```
Enter "exit" or "quit" to logout
Enter "help" or "help command" to show help docs
Input your command:
-->ls -limit=3 -s
obs://bucket-001
obs://bucket-002
obs://bucket-003
Bucket number is: 3

Input your command:
-->
```

- If you use SSH to remotely log in to the Linux or macOS for running obsutil commands, you are advised to set **TMOUT=0** to prevent the program from exiting due to the expiration of the SSH session.

2 Getting Started

2.1 Creating Access Keys (AK and SK)

This section describes how to create access keys (AKs and SKs) in OBS Console. A pair of AK and SK is used to encrypt the signature of a request, ensuring that the request is secure and integral, and that identities of the request sender and receiver are correct.

Scenarios

AKs and SKs support the authentication mechanism of Identity and Access Management (IAM).

- Access key ID (AK): indicates the ID of the access key, which is a unique identifier used together with a secret access key to sign requests cryptographically.
- Secret access key (SK): indicates the private key used together with its associated AK to cryptographically sign requests. The AK and SK are used together to identify a request sender to prevent the request from being modified.

Restrictions and Limitations

Each user can create up to two valid AK/SK pairs.

Prerequisites

An account has been registered and activated.

Procedure

Step 1 In the upper right corner of the console page, select **My Credential** under the username.

Step 2 On the **My Credentials** page, choose **Access Keys** > **Add Access Key**.

Step 3 In the **Add Access Key** dialog box that is displayed, enter the password and its verification code.

NOTE

- If you have not bound an email address or mobile number, enter only the password.
- If you have bound an email address and a mobile number, you can select the verification by email or mobile phone.

Step 4 Click **OK**.

Step 5 In the **Download Access Key** dialog box that is displayed, click **OK** to save the access keys to your browser's default download path.

NOTE

Keep the access keys properly to prevent information leakage. If you click **Cancel** in the dialog box, the access keys will not be downloaded, and you cannot download then later. Re-create access keys if required.

Step 6 Open the downloaded **credentials.csv** file to obtain the access keys (AK and SK).

NOTE

In the access key file, the value in the **Access Key ID** column is the AK, and the value in the **Secret Access Key** column is the SK.

----End

2.2 Performing Initial Configuration

Before using obsutil, you need to configure the interconnection between obsutil and OBS, including the endpoint and access keys (AK and SK) of OBS. You can use obsutil to perform operations on OBS buckets and objects only after obtaining the OBS authentication.

Prerequisites

- You have downloaded the obsutil software package, or compiled the source code to generate obsutil. For details, see **Open-Source Addresses**.
- You have obtained the enabled regions and endpoints of OBS. For details, see **Regions and Endpoints**.
- You have obtained the access keys (AK and SK). For details about how to obtain access keys, see **2.1 Creating Access Keys (AK and SK)**.

Configuration Method

Run the **config** command to initialize obsutil. For details about the **config** command, see **5.1 Updating a Configuration File**. The following is an example:

- In Windows

```
obsutil config -interactive

Please input your ak:
xxxxxxxxxxxxxxxxxxxxxxxx
Please input your sk:
xxxxxxxxxxxxxxxxxxxxxxxx
Please input your endpoint:
xxxxxxxxxxxxxxxxxxxxxxxx
Please input your token:

Config file url:
  C:\Users\tools\.obsutilconfig

Update config file successfully!
```

- In Linux OS or macOS

```
./obsutil config -interactive

Please input your ak:
xxxxxxxxxxxxxxxxxxxxxxx
Please input your sk:
xxxxxxxxxxxxxxxxxxxxxxx
Please input your endpoint:
xxxxxxxxxxxxxxxxxxxxxxx
Please input your token:

Config file url:
  C:\Users\tools\.obsutilconfig

Update config file successfully!
```

NOTE

- After running the preceding commands, a configuration file **.obsutilconfig** is automatically generated in the same home directory of the user who executes obsutil commands (the ~ directory in Linux or macOS, and the **C:\Users**\<*Username*> directory in Windows). **.obsutilconfig** contains all the configuration information of obsutil.
- For details about the parameters in the **.obsutilconfig** file, see **A Parameter Description**.
- The **.obsutilconfig** file contains the AK and SK information of a user. Therefore, it is hidden by default to prevent key disclosure. To query the file, run the following command in the home directory of the user who executes obsutil commands.
  - In Windows

    ```
    dir
    ```

  - In Linux or macOS

    ```
    ls -a
    ```

    or

    ```
    ls -al
    ```

- obsutil encrypts the AK and SK in the **.obsutilconfig** file to ensure key security.

Checking the Connectivity

After the configuration is complete, you can check whether it is correct by running the following commands:

- In Windows

```
obsutil ls -s
```

- In Linux or macOS

```
./obsutil ls -s
```

Check the configuration result based on the command output:

- If the command output contains **Bucket number is:**, the configuration is correct.
- If the command output contains **Http status [403]**, the access keys are incorrectly configured.
- If the command output contains **A connection attempt failed**, then OBS cannot be accessed. In this case, check the network condition.

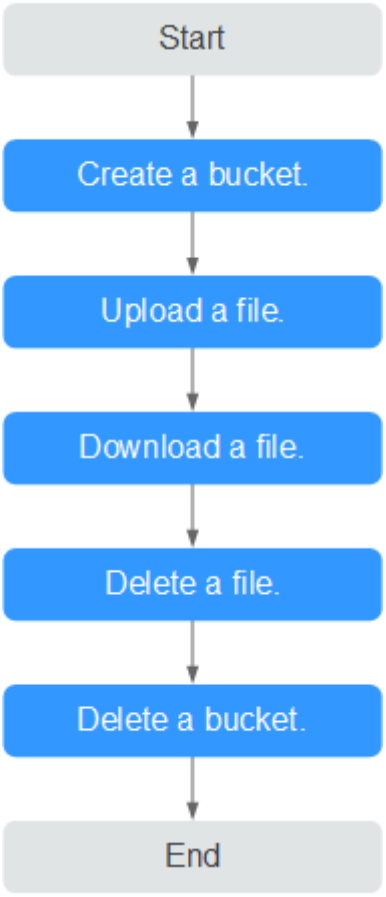
NOTE

If the command output contains **Http status [403]**, you may not have the required permissions for obtaining the bucket list. In this case, further locate the root cause based on the specific situation.

2.3 Quick Start

This section uses the Linux OS as an example to describe how to use obsutil to perform basic data access operations in OBS. For details, see **Figure 2-1**.

Figure 2-1 obsutil flow for a quick start



Prerequisites

- You have obtained obsutil and completed initial configuration.
- You have accessed to the directory where the tool is located.

Procedure

Step 1 Run the **./obsutil mb obs://bucket-test** command to create a new bucket named **bucket-test**.

```
./obsutil mb obs://bucket-test

Create bucket [bucket-test] successfully!
```

Step 2 Run the **./obsutil cp /temp/test.txt obs://bucket-test/test.txt** command to upload the **test.txt** file to bucket **bucket-test**.

```
./obsutil cp /temp/test.txt obs://bucket-test/test.txt

Parallel:       5                   Jobs:           5
Threshold:      52428800            PartSize:       5242880
```

```
Exclude:                            Include:
VerifyLength:  false                VerifyMd5:     false
CheckpointDir: /temp/.obsutil_checkpoint

test.txt:[============================================] 100.00% 48.47 KB/s 0s
Upload successfully, 4.44KB, /temp/test.txt --> obs://bucket-test1/test.txt
```

Step 3 Run the **./obsutil cp obs://bucket-test/test.txt /temp/test1.txt** command to download **test.txt** from bucket **bucket-test** to a local PC.

```
./obsutil cp obs://bucket-test/test.txt /temp/test1.txt

Parallel:      5                    Jobs:          5
Threshold:     52428800             PartSize:      5242880
Exclude:                            Include:
VerifyLength:  false                VerifyMd5:     false
CheckpointDir: /temp/.obsutil_checkpoint

test.txt:[============================================] 100.00% 775.52 KB/s 0s
Download successfully, 4.44KB, obs://bucket-test1/test.txt --> /temp/test1.txt
```

Step 4 Run the **./obsutil rm obs://bucket-test/test.txt -f** command to delete object **test.txt** from bucket **bucket-test**.

```
./obsutil rm obs://bucket-test/test.txt -f

Delete object [test.txt] in the bucket [bucket-test] successfully!
```

Step 5 Run the **./obsutil rm obs://bucket-test -f** command to delete bucket **bucket-test**.

```
./obsutil rm obs://bucket-test -f

Delete bucket [bucket-test] successfully!
```

----End

3 Bucket Commands

3.1 Creating a Bucket

Function

You can use this command to create a bucket. A bucket name must be unique in OBS. One account can create a maximum of 100 buckets.

NOTE

If you create a bucket and name it the same as an existing one in the same account and region, no error will be reported and status code 200 is returned. The bucket properties comply with those set in the first creation request. In other cases, creating a namesake bucket will receive the status code 409, indicating that the bucket already exists.

Command Line Structure

- In Windows
  ```
  obsutil mb obs://bucket [-acl=xxx] [-sc=xxx] [-location=xxx] [-config=xxx]
  ```
- In Linux or macOS
  ```
  ./obsutil mb obs://bucket [-acl=xxx] [-sc=xxx] [-location=xxx] [-config=xxx]
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name **NOTE** A bucket name must comply with the following rules: ● Contains 3 to 63 characters, including lowercase letters, digits, hyphens (-), and periods (.), and starts with a digit or letter. ● Cannot be an IP address. ● Cannot start or end with a hyphen (-) or period (.). ● Cannot contain two consecutive periods (.), for example, **my..bucket**. ● Cannot contain periods (.) and hyphens (-) adjacent to each other, for example, **my-.bucket** or **my.-bucket**.
acl	Optional (additional parameter)	Access control policies that can be specified when creating a bucket. Possible values are: ● private ● public-read ● public-read-write **NOTE** The preceding three values indicate private read and write, public read, and public read and write.
sc	Optional (additional parameter)	Default bucket storage class that can be specified when creating a bucket. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.
location	Mandatory unless the region where the OBS service resides is not the default region (additional parameter)	Region where the bucket resides.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil mb obs://bucket-test** command to create a bucket. The creation is successful.

```
obsutil mb obs://bucket-test

Create bucket [bucket-test] successfully, request id
[0000016979E1D2EA860BB5E80A6B8FCC]
```

- Take the Windows OS as an example. Run the **obsutil mb obs://bucket001** command to create a namesake bucket. The creation fails.

```
obsutil mb obs://bucket001

Create bucket [bucket001] failed, http status [409], error code
[BucketAlreadyExists], error message [The requested bucket name is not available.
The bucket namespace is shared by all users of the system. Please select a
different name andtry again.], request id [04030000016757F31A0333281A6B1E92]
```

3.2 Listing Buckets

Function

You can use this command to obtain the bucket list. In the list, bucket names are displayed in lexicographical order.

Command Line Structure

- In Windows
  ```
  obsutil ls [-s] [-sc] [-j=1] [-limit=1] [-config=xxx]
  ```
- In Linux or macOS
  ```
  ./obsutil ls [-s] [-sc] [-j=1] [-limit=1] [-config=xxx]
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
s	Optional (additional parameter)	Displays simplified query result. **NOTE** In the simplified format, the returned result contains only the bucket name.
sc	Optional (additional parameter)	Queries the storage classes of the buckets when listing buckets.
j	Optional (additional parameter). It must be used together with **sc**.	Indicates the maximum number of concurrent tasks for querying the bucket storage class. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
limit	Optional (additional parameter)	Maximum number of buckets that can be queried. If the value is less than 0, all buckets are listed. If it is left blank, a maximum of 1000 buckets can be listed by default.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil ls -limit=5** command to obtain the bucket list.

```
obsutil ls -limit=5

Bucket                          CreationDate                Location
BucketType
obs://bucket001                 2018-09-03T01:53:02Z        example
OBJECT

obs://bucket002                 2018-11-01T01:40:01Z        example
OBJECT

obs://bucket003                 2018-10-25T11:45:45Z        example
OBJECT

obs://bucket004                 2018-10-26T02:33:09Z        example
OBJECT

obs://bucket005                 2018-10-26T02:34:50Z        example
OBJECT

Bucket number is: 5
```

NOTE

In the bucket listing result, the **BucketType** field indicates the bucket type: **OBJECT** indicates the bucket for object storage; **POSIX** indicates the parallel file system.

3.3 Querying Bucket Properties

Function

You can use this command to query the basic properties of a bucket, including its default storage class, region, version ID, storage usage, bucket quota, and the number of objects in the bucket.

Command Line Structure

- In Windows

```
obsutil stat obs://bucket [-acl] [-config=xxx]
```

- In Linux or macOS

```
./obsutil stat obs://bucket [-acl] [-config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
acl	Optional	Queries the access control policies of the bucket while querying bucket properties.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Field	Description
Bucket	Bucket name
StorageClass	Default storage class of the bucket
Location	Region where the bucket resides
ObsVersion	Version of the bucket
BucketType	Type of a bucket. **OBJECT** indicates a bucket for object storage. **POSIX** indicates a bucket used as a parallel file system.
ObjectNumber	Number of objects in the bucket
Size	Storage usage of the bucket, in bytes
Quota	Bucket quota. Value **0** indicates that no upper limit is set for the bucket quota.
Acl	Access control policy of the bucket

Running Example

- Take the Windows OS as an example. Run the **obsutil stat obs://bucket-test** command to query the basic properties of bucket **bucket-test**.

```
obsutil stat obs://bucket-test

Bucket:
  obs://bucket-test
StorageClass:
  standard
Location:
  southchina
ObsVersion:
  3.0
ObjectNumber:
  8005
Size:
  320076506
```

```
Quota:
  0
```

3.4 Setting Bucket Properties

Function

You can use this command to set the properties of a bucket, such as storage classes and access policies.

Command Line Structure

- In Windows
  ```
  obsutil chattri obs://bucket [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-config=xxx]
  ```
- In Linux or macOS
  ```
  ./obsutil chattri obs://bucket [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-
  config=xxx]
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
sc	Optional (additional parameter)	Default storage class of the bucket. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.
acl	Optional (additional parameter)	Access control policies that can be specified for buckets. Possible values are: ● private ● public-read ● public-read-write **NOTE** The preceding three values indicate private read and write, public read, and public read and write.

Parameter	Optional or Mandatory	Description

aclXml	Optional (additional parameter)	Access control policy of the bucket, in XML format.
		`<AccessControlPolicy>`
		`  <Owner>`
		`  <ID>ownerid</ID>`
		`  </Owner>`
		`  <AccessControlList>`
		`  <Grant>`
		`  <Grantee>`
		`  <ID>userid</ID>`
		`  </Grantee>`
		`  <Permission>[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]</Permission>`
		`  </Grant>`
		`  <Grant>`
		`  <Grantee>`
		`  <Canned>Everyone</Canned>`
		`  </Grantee>`
		`  <Permission>[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]</Permission>`
		`  </Grant>`
		`  </AccessControlList>`
		`</AccessControlPolicy>`
		NOTE
		● **Owner**: Optional. Specify the bucket owner's ID.
		● In **AccessControlList**, the **Grant** field contains the authorized users. **Grantee** specifies the IDs of authorized users. **Canned** specifies the authorized user group (currently, only **Everyone** is supported).
		● The following permissions can be granted: WRITE (write), WRITE_ACP (write ACL), READ (read), READ_ACP (read ACL), and FULL_CONTROL (full control).
		NOTICE
		Because angle brackets (<) and (>) are unavoidably included in the parameter value, you must use quotation marks to enclose them for escaping when running the command. Use single quotation marks for Linux or macOS and quotation marks for Windows.

config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

NOTE

Only **sc**, **acl**, or **aclXml** can be set for each command.

Running Example

- Take the Windows OS as an example. Run the **obsutil chattri obs://bucket-test -acl=private** command to change the access control policy of the bucket to private read and write.

```
obsutil chattri obs://bucket-test -acl=private
```

```
Set the acl of bucket [bucket-test] to [private] successfully, request id
[04050000016836C5DA6FB21F14A2A0C0]
```

3.5 Deleting a Bucket

Function

You can use this command to delete a bucket. The bucket to be deleted must be empty (containing no objects, historical versions, or fragments).

NOTE

To delete a non-empty bucket, run the commands in **4.19 Deleting a Multipart Upload Task** and in **4.10 Deleting an Object** to clear the bucket, and then run the following command to delete the bucket.

Command Line Structure

- In Windows
  ```
  obsutil rm obs://bucket [-f] [-config=xxx]
  ```
- In Linux or macOS
  ```
  ./obsutil rm obs://bucket [-f] [-config=xxx]
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
f	Optional (additional parameter)	Runs in force mode.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil rm obs://bucket-test** command to delete bucket **bucket-test**.

```
obsutil rm obs://bucket-test

Do you want delete bucket [bucket-test] ? Please input (y/n) to confirm:
y
Delete bucket [bucket-test] successfully!
```

4 Object Commands

4.1 Creating a Folder

Function

You can use this command to create a folder in a specified bucket or local file system.

Command Line Structure

- In Windows
  - Creating a folder in a specified bucket
    ```
    obsutil mkdir obs://bucket/folder[/subfolder1/subfolder2] [-config=xxx]
    ```
  - Creating a folder in the local file system
    ```
    obsutil mkdir folder_url [-config=xxx]
    ```
- In Linux or macOS
  - Creating a folder in a specified bucket
    ```
    ./obsutil mkdir obs://bucket/folder[/subfolder1/subfolder2] [-config=xxx]
    ```
  - Creating a folder in the local file system
    ```
    ./obsutil mkdir folder_url [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory when creating a folder in a specified bucket	Bucket name
folder	Mandatory when creating a folder in a specified bucket	Folder path in the bucket. This value can contain multi-level folders. Separate each level with a slash (/).

Parameter	Optional or Mandatory	Description
folder_url	Mandatory when creating a folder in the local file system	Folder path in the local file system. The value can be an absolute path or a relative path.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil mkdir obs://bucket-test/folder1/folder2** command to create a folder in a bucket.

```
obsutil mkdir obs://bucket-test/folder1/folder2

Create folder [obs://bucket-test/folder1/] successfully, request id
[0000016979E1D23C860BB3D8E4577C5E]
Create folder [obs://bucket-test/folder1/folder2] successfully, request id
[0000016979E1D2B2860BB5181229C72C]
```

4.2 Uploading an Object

Function

You can use this command to upload one or more local files or folders to a specified path in OBS. These files can be texts, images, videos, or any other type of files.

NOTICE

Do not change the local file or folder when uploading it. Otherwise, the upload may fail or data may be inconsistent.

Restrictions

obsutil has restrictions on the size of files or folders to be uploaded. You can upload an empty file or folder of 0 bytes. You can also upload a single file or folder with a maximum size of 5 GB in normal mode or a single file or folder with a maximum size of 48.8 TB in multipart mode.

Command Line Structure

- In Windows
  - Uploading a file

```
obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-
u] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-
meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-o=xxx] [-
config=xxx]
```

  - Uploading a folder
    ```
    obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```
  - Uploading multiple files/folders
    ```
    obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```

- In Linux or macOS
  - Uploading a file
    ```
    ./obsutil cp file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Uploading a folder
    ```
    ./obsutil cp folder_url obs://bucket[/key] -r [-arcDir=xxx] [-dryRun] [-link] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```
  - Uploading multiple files/folders
    ```
    ./obsutil cp file1_url,folder1_url|filelist_url obs://bucket[/prefix] -msm=1 [-r] [-arcDir=xxx] [-dryRun] [-link] [-f] [-u] [-vlength] [-vmd5] [-flat] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
file_url	Optional for uploading multiple files/folders Mandatory for uploading a file	Local file path **NOTE** ● No paths can be nested when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple file paths, for example, **file_url1,file_url2**. ● Files and folders can both be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.

Parameter	Optional or Mandatory	Description
folder_url	Optional for uploading multiple files/folders Mandatory for uploading a folder	Local folder path **NOTE** ● If **flat** is not configured when uploading a folder, the entire folder is uploaded. If **flat** is configured, all files in the folder are uploaded. ● No paths can be nested when uploading multiple files/folders. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time. ● If this parameter is configured when uploading multiple files/folders, **msm** must be set to **1**. In this case, use commas (,) to separate multiple folder paths, for example, **folder_url1,folder_url2**. ● Files and folders can be included when uploading multiple files/folders. For example, **file_url1,folder_url1,file_url2,folder_url2**.
filelist_url	Optional for uploading multiple files/folders	Indicates the path of the file that contains the list of files/folders to be uploaded. If this parameter is configured, **msm** must be set to **2**. **NOTE** ● The list file is in common text file formats, such as TXT and CSV. Each line in the file indicates a file or folder to be uploaded. For example: file_url1 file_url2 folder_url1 folder_url2 ● No paths can be nested in the list file. For example, you cannot specify **/a/b/c** and **/a/b/** at the same time.
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Optional	Indicates the object name or object name prefix specified when uploading a file, or the object name prefix specified when uploading a folder. The rules are as follows: ● If this parameter is left blank when uploading a file, the file is uploaded to the root directory of the bucket and the object name is the file name. If the value ends with a slash (/), the value is used as the object name prefix when the file is uploaded, and the object name is the value plus the file name. If the value does not end with a slash (/), the file is uploaded with the value as the object name. ● If this parameter is left blank when uploading a folder, the folder is uploaded to the root directory of the bucket. If the value ends with a slash (/), the value is used as the object name prefix of the folder to be uploaded. If the value does not end with a slash (/), the folder to be uploaded is prefixed with the value plus a slash (/). **NOTE** For details about how to use this parameter, see **6.1 Upload Examples**.
fr	Optional for uploading a file (additional parameter)	Generates an operation result list when uploading a file.
flat	Optional for uploading a folder or multiple files/folders (additional parameter)	Uploads all files in a folder but not the folder itself.
arcDir	Optional (additional parameter)	Path to which the uploaded files are archived
dryRun	Optional (additional parameter)	Conducts a dry run.

Parameter	Optional or Mandatory	Description
link	Optional (additional parameter)	Uploads the actual path of the symbolic-link file/ folder **NOTICE** ● If this parameter is not specified and the file to be uploaded is a symbolic-link file whose target file does not exist, the exception message "The system cannot find the file specified" will be displayed in Windows OS, while the exception message "No such file or directory" will be displayed in macOS or Linux OS. ● Avoid the symbolic link loop of a folder, otherwise, the upload will exit due to panic. If you do not want the system to panic, set **panicForSymbolicLinkCircle** to **false** in the configuration file.
u	Optional (additional parameter)	Indicates incremental upload. If this parameter is set, each file can be uploaded only when it does not exist in the bucket, its size is different from the namesake one in the bucket, or it has the latest modification time.
vlength	Optional (additional parameter)	After the upload is complete, check whether the sizes of the objects in the bucket are the same as those of the local files.
vmd5	Optional (additional parameter)	After the upload completes, check whether the MD5 values of the objects in the bucket are the same as those of the local files. **NOTE** ● If the size of the file or folder to be uploaded is too large, using this parameter will degrade the overall performance due to MD5 calculation. ● After the MD5 value verification is successful, the parameter value is set to the object metadata **x-obs-md5chksum**, which is used for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart upload tasks when uploading a file. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart upload, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the file or folder to be uploaded is smaller than the threshold, upload it directly. Otherwise, a multipart upload is required. ● If you upload a file or folder directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
acl	Optional (additional parameter)	Access control policies that can be specified when uploading files. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.
sc	Optional (additional parameter)	Indicates the storage classes of objects that can be specified when uploading files. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.
meta	Optional (additional parameter)	Indicates the customized metadata that can be specified when uploading files. The format is **key1:value1#key2:value2#key3:value3**. **NOTE** The preceding value indicates that the objects in the bucket contain three groups of customized metadata after the file is uploaded: **key1:value1**, **key2:value2**, and **key3:value3**.

Parameter	Optional or Mandatory	Description
ps	Optional (additional parameter)	Indicates the size of each part in a multipart upload task, in bytes. The value ranges from 100 KB to 5 GB. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source file size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart upload and saved to the **upload** subfolder. After the upload succeeds, its part record is deleted automatically. If the upload fails or is suspended, the system attempts to resume the task according to its part record when you perform the upload the next time.
r	Mandatory for uploading a folder (additional parameter) Optional for uploading multiple files/folders	Indicates files and subfolders within the folder when uploading a folder recursively.
f	Optional for uploading a folder or multiple files/folders (additional parameter)	Runs in force mode.
j	Optional for uploading a folder or multiple files/folders (additional parameter)	Indicates the maximum number of concurrent tasks for uploading a folder. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
msm	Mandatory for uploading multiple files/folders (additional parameter)	Enables the mode for uploading multiple files/folders. Possible values are **1** and **2**. **NOTE** ● If **msm** is set to **1**, the source URL indicates a list of file/folder names separated by commas. ● If **msm** is set to **2**, the source URL indicates a file containing a list of file/folder names. ● If the file or folder name already contains commas (,), do not set **msm** to **1**. ● If parameter **r** is not set, the folders in the list will not be uploaded.
exclude	Optional for uploading a folder or multiple files/folders (additional parameter)	Indicates the file matching patterns that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the file to be uploaded matches the value of this parameter, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder.

Parameter	Optional or Mandatory	Description
include	Optional for uploading a folder or multiple files/folders (additional parameter)	Indicates the file matching patterns that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be uploaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is uploaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder.
timeRange	Optional for uploading a folder or multiple files/folders (additional parameter)	Indicates the time range matching pattern when uploading files. Only files whose last modification time is within the configured time range are uploaded. This pattern has a lower priority than the file matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured file matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*****, all files whose last modification time is later than *time1* are matched. **NOTICE** Time in the matching pattern is the UTC time.
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on folders.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time*_**TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Field	Description
Parallel	Parameter **-p** in the request
Jobs	Parameter **-j** in the request
Threshold	Parameter **-threshold** in the request
PartSize	Parameter **-ps** in the request
Exclude	Parameter **-exclude** in the request
Include	Parameter **-include** in the request
TimeRange	Parameter **-timeRange** in the request
VerifyLength	Parameter **-vlength** in the request
VerifyMd5	Parameter **-vmd5** in the request
CheckpointDir	Parameter **-cpd** in the request
OutputDir	Parameter **-o** in the request
ArcDir	Parameter **-arcDir** in the request
Succeed count	Number of successful tasks
Failed count	Number of failed tasks

Field	Description
Skip count	Number of tasks that are skipped during incremental upload, download, or copy, and synchronous upload, download, or copy. **NOTE** Skipped tasks are recorded into successful tasks.
Warning count	Number of tasks that are executed successfully but contain warnings. **NOTE** ● The task for which a warning is generated may be a failure or a success, which needs to be further determined according to the corresponding result list. ● The number of tasks that generate warnings is independent of the number of successful or failed tasks. The total number of tasks is the number of successful tasks plus the number of failed tasks.
Succeed bytes	Number of bytes that are successfully uploaded or downloaded.
max cost	Maximum duration of all tasks, in ms
min cost	Minimum duration of all tasks, in ms
average cost	Average duration of all tasks, in ms
average tps	The average number of tasks completed per second
Task id	Unique ID of an operation, which is used to search for the result list generated in a batch task

Running Examples

- Take the Windows OS as an example. Run the **obsutil cp d:\temp\test.txt obs://bucket-test/key** command to upload the **test.txt** file in the **temp** directory in the **D:** drive to bucket **bucket-test** and rename the file as **key**.

```
obsutil cp d:\temp\test.txt obs://bucket-test/key

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

[=====================================================] 100.00% 1.68 MB/s 5s
Upload successfully, 8.46MB, d:\temp\test.txt --> obs://bucket-test/key
```

- Take the Windows OS as an example. Run the **obsutil cp d:\temp obs://bucket-test -f -r** command to recursively upload all files and subfolders in the **temp** directory in the **D:** drive to the **temp** folder in bucket **bucket-test**.

```
obsutil cp d:\temp obs://bucket-test -f -r

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx

[=====================================================] 100.00% 2.02 KB/s 0s
Succeed count is:   5         Failed count is:    0
```

```
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71]
Task id is: 104786c8-27c2-48fc-bc6a-5886596fb0ed
```

- For more examples, see **6.1 Upload Examples**.

4.3 Querying Object Properties

Function

You can use this command to query the basic properties of an object.

Command Line Structure

- In Windows
  ```
  obsutil stat obs://bucket/key [-acl] [-config=xxx]
  ```
- In Linux or macOS
  ```
  ./obsutil stat obs://bucket/key [-acl] [-config=xxx]
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Mandatory	Object name
acl	Optional	Queries the access control policies of the object at the same time.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Field	Description
Key	Object name
LastModified	Last modification time of the object
Size	Object size, in bytes
StorageClass	Storage class of the object
MD5	Real MD5 of the object **NOTE** You can query this value only after running the **cp** command and configuring the **-vmd5** parameter.
ETag	ETag value of an object calculated on the server
ContentType	Content-Type of the object

Field	Description
Metadata	Customized metadata of the object
Acl	Access control policy of the object

Running Example

- Take the Windows OS as an example. Run the **obsutil stat obs://bucket-test/key** command to query the basic properties of an object.

```
obsutil stat obs://bucket-test/key

Key:
  obs://bucket-test/key
LastModified:
  2018-11-16T02:15:49Z
Size:
  7
ETag:
  43d93b553855b0e1fc67e31c28c07b65
ContentType:
  text/plain
```

4.4 Setting Object Properties

Function

You can use this command to set properties of an object or set properties of objects in batches by a specified object name prefix.

NOTE

You can set storage classes only for buckets whose version is 3.0.

Command Line Structure

- In Windows
  - Setting properties of a single object
    ```
    obsutil chattri obs://bucket/key [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Setting properties of objects in batches
    ```
    obsutil chattri obs://bucket[/key] -r [-f] [-v] [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx]
    ```
- In Linux or macOS
  - Setting properties of a single object
    ```
    ./obsutil chattri obs://bucket/key [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Setting properties of objects in batches
    ```
    ./obsutil chattri obs://bucket[/key] -r [-f] [-v] [-sc=xxx] [-acl=xxx] [-aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Mandatory for setting properties of an object. Optional for setting properties of objects in batches.	Indicates the name of the object whose properties are to be set, or the name prefix of objects whose properties are to be set in batches. **NOTE** If this parameter is left blank during batch operation, properties of all objects in the bucket are set.
sc	Optional (additional parameter)	Storage classes of objects. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data. **NOTE** For an object whose storage class is **cold**, restore the object first and then set its storage class. To restore an object, see **4.14 Restoring Objects from OBS Cold**.
acl	Optional (additional parameter)	Access control policies that can be specified for objects. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.

Parameter	Optional or Mandatory	Description

aclXml	Optional (additional parameter)	Access control policy of the bucket, in XML format.
		`<AccessControlPolicy>`
		`  <Owner>`
		`  <ID>ownerid</ID>`
		`  </Owner>`
		`  <AccessControlList>`
		`  <Grant>`
		`  <Grantee>`
		`  <ID>userid</ID>`
		`  </Grantee>`
		`  <Permission>[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]</Permission>`
		`  </Grant>`
		`  <Grant>`
		`  <Grantee>`
		`  <Canned>Everyone</Canned>`
		`  </Grantee>`
		`  <Permission>[WRITE|WRITE_ACP|READ|READ_ACP|FULL_CONTROL]</Permission>`
		`  </Grant>`
		`  </AccessControlList>`
		`</AccessControlPolicy>`
		NOTE
		● **Owner**: Optional. Specify the object owner's ID.
		● In **AccessControlList**, the **Grant** field contains the authorized users. **Grantee** specifies the IDs of authorized users. **Canned** specifies the authorized user group (currently, only **Everyone** is supported).
		● The following permissions can be granted: WRITE (write), WRITE_ACP (write ACL), READ (read), READ_ACP (read ACL), and FULL_CONTROL (full control).
		NOTICE
		Because angle brackets (<) and (>) are unavoidably included in the parameter value, you must use quotation marks to enclose them for escaping when running the command. Use single quotation marks for Linux or macOS and quotation marks for Windows.

versionId	Optional for setting properties of an object (additional parameter)	Version ID of the object whose properties are to be set
fr	Optional for setting properties of an object (additional parameter)	Generates an operation result list when setting properties of an object.
f	Optional when setting properties of objects in batches (additional parameter)	Runs in force mode.

Parameter	Optional or Mandatory	Description
r	Mandatory when setting properties of objects in batches (additional parameter)	Sets properties of objects in batches based on a specified object name prefix.
v	Optional when setting properties of objects in batches (additional parameter)	Sets properties of versions of objects in batches based on a specified object name prefix.
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **chattri_{succeed | failed}_report_***time*_**TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
j	Optional when setting properties of objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for setting object properties in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

NOTE

Only one of **sc**, **acl**, and **aclXml** can be set for each command.

Response

Refer to **Response** for uploading an object.

Running Examples

- Take the Windows OS as an example, run the **obsutil chattri obs://bucket-test/key -acl=public-read** command to set the access permission to an object to public read.

```
obsutil chattri obs://bucket-test/key -acl=public-read
```

```
Set the acl of object [key] in the bucket [bucket-test] to [public-read]
successfully, request id [04050000016836DDFA73B2B5320E2651]
```

- Take the Windows OS as an example, run the **obsutil chattri obs://bucket-test -r -f -acl=public-read** command to set the access permission to all objects in the bucket to public read.

```
obsutil chattri obs://bucket-test -r -f -acl=public-read

[----------------------------------------------] 100.00% tps:155.15 5/5 233ms
Succeed count is:   5         Failed count is:    0
Metrics [max cost:177 ms, min cost:53 ms, average cost:102.40 ms, average tps:
20.41]
Task id is: 9d7f73ff-f747-4fdd-9b2a-815ba2dc3b07
```

4.5 Listing Objects

Function

You can use this command to query objects or object versions in a bucket. All objects are listed in lexicographical order by object name and version ID.

Command Line Structure

- In Windows
  ```
  obsutil ls obs://bucket[/prefix] [-s] [-d] [-v] [-marker=xxx] [-
  versionIdMarker=xxx] [-bf=xxx] [-limit=1] [-config=xxx]
  ```
- In Linux or macOS
  ```
  ./obsutil ls obs://bucket[/prefix] [-s] [-d] [-v] [-marker=xxx] [-
  versionIdMarker=xxx] [-bf=xxx] [-limit=1] [-config=xxx]
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
prefix	Optional	Prefix of an object name for listing objects **NOTE** If this parameter is left blank, all objects in the bucket are listed.
s	Optional (additional parameter)	Displays simplified query result. **NOTE** In the simplified format, the returned result contains only the object name.
d	Optional (additional parameter)	Lists only objects and subdirectories in the current directory, instead of recursively listing all objects and subdirectories. **NOTE** According to the naming conventions in OBS, a slash (/) is used as the directory separator.

Parameter	Optional or Mandatory	Description
v	Optional (additional parameter)	Lists versions of an object in a bucket. The result contains the latest version and historical versions (if any) of the object.
marker	Optional (additional parameter)	Object name to start with when listing objects in a bucket. All objects are listed in lexicographical order by object name. **NOTE** For details about how to use this parameter, see **6.7 List Examples**.
versionIdMarker	Optional (additional parameter). It must be used together with the **v** and **marker** parameters.	Version ID to start with when listing versions of objects in a bucket. All versions and objects are listed in lexicographical order by object name and version ID. **NOTE** If the value of **versionIdMarker** is not a version ID specified by **marker**, **versionIdMarker** is invalid.
bf	Optional (additional parameter)	Display formats of bytes in the listing result. Possible values are: ● human-readable ● raw **NOTE** If this parameter is not configured, the display format of bytes in the result is determined by the **humanReadableFormat** parameter in the configuration file.
limit	Optional (additional parameter)	Maximum number of objects that can be listed. If the value is less than or equal to 0, all objects are listed. If it is left blank, 1000 objects are listed by default. **NOTE** If there are a large number of objects in a bucket, you are advised to set this parameter to limit the number of objects to be listed each time. If not all objects are listed, **marker** and **versionIdMarker** of the next request will be returned in the result, which you can use to list the remaining objects.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil ls obs://bucket-test -limit=10** command to list objects in the bucket.

```
obsutil ls obs://bucket-test -limit=10
```

```
Folder list:
obs://bucket-test/api/

Object list:
key                                                 LastModified
Size      StorageClass        ETag
obs://bucket-test/AUTHORS                           2018-11-16T02:15:49Z
33243     standard            "796393c1eaf502ef56a85c2ceb640aea"

obs://bucket-test/CONTRIBUTING.md                   2018-11-16T02:15:49Z
1366      standard            "12d93325ba6131f852daecd18dd65edc"

obs://bucket-test/CONTRIBUTORS                      2018-11-16T02:15:49Z
45710     standard            "b486b5003e6215c9199e86ab3ccec9fa"

obs://bucket-test/LICENSE                           2018-11-16T02:15:49Z
1479      standard            "5d4950ecb7b26d2c5e4e7b4e0dd74707"

obs://bucket-test/PATENTS                           2018-11-16T02:15:49Z
1303      standard            "3a55d95595a6f9e37dee53826b4daff2"

obs://bucket-test/README.md                         2018-11-16T02:15:49Z
1399      standard            "97351fd7946b9ea021a31a86ba2a10ab"

obs://bucket-test/VERSION                           2018-11-16T02:15:49Z
7         standard            "43d93b553855b0e1fc67e31c28c07b65"

obs://bucket-test/api/README                        2018-11-16T02:15:49Z
521       standard            "4e9e63a87075df60cdf65c8ce9e92117"

obs://bucket-test/api/except.txt                    2018-11-16T02:15:49Z
20194     standard            "8eb96de3f60447e2f09a7531c99fb3ee"

Next marker is: api/except.txt
Folder number is: 1
File number is: 9
```

- For more examples, see **6.7 List Examples**.

4.6 Copying an Object

Function

You can use this command to copy a single object or copy objects in batches by a specified object name prefix.

NOTICE

- Do not change the source objects in the OBS bucket when copying a single object or objects in batches. Otherwise, the operation may fail or data may be inconsistent.
- If the storage class of the object to be copied is **cold**, you must restore the object to be copied first. Otherwise, the copy fails.
- To copy objects, you must have the read permission on the objects to be copied and the write permission on the destination bucket.
- If the client-side cross-region replication function is not enabled, ensure that the source bucket and destination bucket are in the same region.

Command Line Structure

- In Windows
  - Copying a single object
    ```
    obsutil cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun][-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Copying objects in batches
    ```
    obsutil cp obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun][-f] [-flat] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```
- In Linux or macOS
  - Copying a single object
    ```
    ./obsutil cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Copying objects in batches
    ```
    ./obsutil cp obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun] [-f] [-flat] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```

NOTE

- The source path and destination path cannot be the same.
- The source path and destination path cannot be partly overlapped either. If the source path overlaps with the prefix of the destination path, recursive replication applies. If the destination path overlaps with the prefix of the source path, the replication may overwrite objects in the source path.

Parameter Description

Parameter	Optional or Mandatory	Description
srcbucket	Mandatory	Source bucket name
dstbucket	Mandatory	Destination bucket name
dest	Optional	Indicates the destination object name when copying an object, or the name prefix of destination objects when copying objects in batches.

Parameter	Optional or Mandatory	Description
key	Mandatory for copying an object. Optional for copying objects in batches.	Indicates the source object name when copying an object, or the name prefix of source objects when copying objects in batches. The rules are as follows: ● This parameter cannot be left blank when copying an object. If **dest** is left blank, the source object is copied to the root directory of the destination bucket. If the value of **dest** ends with a slash (/), the destination object name is the value of **dest** plus the source object name. Otherwise, the destination object name is the value of **dest**. ● If this parameter is left blank when copying objects in batches, all objects in the source bucket are copied. If not, objects whose name prefix is the set value in the source bucket are copied. The rules for confirming the name of the destination object are as follows: – If the value of **dest** ends with a slash (/), the destination object name is the value of **dest** plus the source object name. – If the value of **dest** does not end with a slash (/), the destination object name is *the value of **dest**/ source object name*. **NOTE** ● If this parameter is configured but the **flat** parameter is not when copying objects in batches, the name of the source object contains the name prefix of the parent object. If **flat** is configured, then the name of the source object does not contain the name prefix of the parent object. ● For details about how to use this parameter, see **6.5 Copy Examples**.
fr	Optional for copying an object (additional parameter)	Generates an operation result list when copying an object.
flat	Optional for copying objects in batches (additional parameter)	The name prefix of the parent object is excluded when copying objects in batches.
dryRun	Optional (additional parameter)	Conducts a dry run.

Parameter	Optional or Mandatory	Description
crr	Optional (additional parameter)	Enables the client-side cross-region replication function. In this mode, data is directly copied to the destination bucket from the source bucket through data stream. The buckets can by any two OBS buckets. **NOTE** ● If this parameter is configured, ensure that the configuration of client-side cross-region replication is updated in the configuration file. For details, see **5.1 Updating a Configuration File**. ● The configurations of the source bucket and destination bucket are respectively **akCrr/skCrr/tokenCrr/endpointCrr** and **ak/sk/token/endpoint** in the configuration file. **NOTICE** After this function is enabled, both upload and download bandwidth are occupied.
vlength	Optional (additional parameter)	Verifies whether the object size in the destination bucket is the same as that in the source bucket after the copy task completes. **NOTE** This parameter must be used together with **crr**.
vmd5	Optional (additional parameter)	Verifies whether the MD5 value of the destination bucket is the same as that of the source bucket after the copy task completes. **NOTE** ● This parameter must be used together with **crr**. ● Objects in the source bucket must contain metadata **x-obs-md5chksum**. Otherwise, MD5 verification will be skipped. After the MD5 value verification is successful, the parameter value is set to the destination object metadata **x-obs-md5chksum**, which is used for later MD5 verification during download or copy.
u	Optional (additional parameter)	Indicates incremental copy. If this parameter is set, each object can be copied only when it does not exist in the destination bucket, its size is different from the namesake one in the destination bucket, or it has the latest modification time.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart copy tasks when copying an object. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart copy, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be copied is smaller than the threshold, copy the object directly. If not, a multipart copy is required. ● If you copy an object directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
versionId	Optional for copying an object (additional parameter)	Source object version ID that can be specified when copying an object
acl	Optional (additional parameter)	Access control policies for destination objects that can be specified when copying objects. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.
sc	Optional (additional parameter)	Storage classes of the destination objects that can be specified when copying objects. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.

Parameter	Optional or Mandatory	Description
meta	Optional (additional parameter)	Metadata of destination objects that can be specified when copying objects. The format is *key1*:*value1*#*key2*:*value2*#*key3*:*value3*. **NOTE** The preceding value indicates that the destination objects in the bucket contain three groups of customized metadata after objects are copied: **key1:value1**, **key2:value2**, and **key3:value3**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart copy task, in bytes. The value ranges from 100 KB to 5 GB. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart copy and saved to the **copy** subfolder. After the copy succeeds, its part record is deleted automatically. If the copy fails or is suspended, the system attempts to resume the task according to its part record when you perform the copy the next time.
r	Mandatory for copying objects in batches (additional parameter)	Copies objects in batches based on a specified name prefix of objects in the source bucket.
f	Optional for copying objects in batches (additional parameter)	Runs in force mode.
j	Optional for copying objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for copying objects in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
exclude	Optional for copying objects in batches (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the object to be copied matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).
include	Optional for copying objects in batches (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be copied does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is copied. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
timeRange	Optional for copying objects in batches (additional parameter)	Indicates the time range matching pattern when copying objects. Only objects whose last modification time is within the configured time range are copied. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*, all files whose last modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Examples

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/key obs://bucket-test2** command to copy a single object.

```
obsutil cp obs://bucket-test/key obs://bucket-test2

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

[=====================================================] 100.00% 6/s 0s
Copy successfully, 19B, obs://bucket-test/key --> obs://bucket-test2/key
ext.txt
```

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/temp/ obs://bucket-test2 -f -r** command to copy objects in batches.

```
obsutil cp obs://bucket-test/temp/ obs://bucket-test2 -r -f

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx


[============================================================] 100.00% 10/s 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:
9.71]
Task id is: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
```

- For more examples, see **6.5 Copy Examples**.

4.7 Moving an Object

Function

You can use this command to move a single object or move objects in batches by a specified object name prefix.

NOTICE

- Do not change the source objects in the OBS bucket when moving objects. Otherwise, the operation may fail or data may be inconsistent.
- The source objects are deleted after the move operation succeeds.

Command Line Structure

- In Windows

  - Moving a single object
    ```
    obsutil mv obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Moving objects in batches
    ```
    obsutil mv obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun] [-f] [-flat] [-u] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```

- In Linux OS or macOS
  - Moving a single object
    ```
    ./obsutil mv obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Moving objects in batches
    ```
    ./obsutil mv obs://srcbucket[/key] obs://dstbucket[/dest] -r [-dryRun] [-f] [-flat] [-u] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```

NOTE

- The source path and destination path cannot be the same.
- The source and destination paths cannot be nested when moving objects in batches.

Parameter Description

Parameter	Optional or Mandatory	Description
srcbucket	Mandatory	Source bucket name
dstbucket	Mandatory	Destination bucket name
dest	Optional	Indicates the destination object name when moving a single object, or the name prefix of destination objects when moving objects in batches.

Parameter	Optional or Mandatory	Description
key	Mandatory for moving a single object Optional for moving objects in batches	Indicates the source object name when moving a single object, or the name prefix of source objects when moving objects in batches. The rules are as follows: ● This parameter cannot be left blank when moving a single object. If **dest** is left blank, the source object is moved to the root directory of the destination bucket. If the value of **dest** ends with a slash (/), the destination object name is the value of **dest** plus the source object name. Otherwise, the destination object name is the value of **dest**. ● If this parameter is left blank when moving objects in batches, all objects in the source bucket are moved. If not, objects whose name prefix is the set value in the source bucket are moved. The rules for confirming the name of the destination object are as follows: – If the value of **dest** ends with a slash (/), the destination object name is the value of **dest** plus the source object name. – If the value of **dest** does not end with a slash (/), the destination object name is *the value of **dest**/ source object name*. **NOTE** ● If this parameter is configured but parameter **flat** is not when moving objects in batches, the name of the source object contains the name prefix of the parent object. If **flat** is configured, then the name of the source object does not contain the name prefix of the parent object. ● For details about how to use this parameter, see **Command Line Structure**.
fr	Optional for moving an object (additional parameter)	Generates an operation result list when moving an object.
flat	Optional for moving objects in batches (additional parameter)	The name prefix of the parent object is excluded when moving objects in batches.
dryRun	Optional (additional parameter)	Conducts a dry run.

Parameter	Optional or Mandatory	Description
u	Optional (additional parameter)	Indicates incremental move. If this parameter is set, each object can be moved only when it does not exist in the destination bucket, its size is different from the namesake one in the destination bucket, or it has the latest modification time. **NOTE** If the size and modification time of the destination object are the same as those of the source object, the source object is directly deleted instead of being moved.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart move tasks when moving an object. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart move, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be moved is smaller than the threshold, move the object directly. If not, a multipart move is required. ● If you move an object directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
versionId	Optional for moving an object (additional parameter)	Source object version ID that can be specified when moving a single object
acl	Optional (additional parameter)	Access control policies for destination objects that can be specified when moving objects. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Storage classes of the destination objects that can be specified when moving objects. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.
meta	Optional (additional parameter)	Metadata of destination objects that can be specified when copying objects. The format is *key1***:***value1***#***key2***:***value2***#***key3***:***value3*. **NOTE** The preceding value indicates that the destination objects in the bucket contain three groups of customized metadata after objects are copied: **key1:value1**, **key2:value2**, and **key3:value3**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart move task, in bytes. The value ranges from 100 KB to 5 GB. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart move and saved to the **copy** subfolder. After the move succeeds, its part record is deleted automatically. If the move fails or is suspended, the system attempts to resume the task according to its part record when you perform the move the next time.
r	Mandatory for moving objects in batches (additional parameter)	Moves objects in batches based on a specified name prefix of objects in the source bucket.

Parameter	Optional or Mandatory	Description
f	Optional for moving objects in batches (additional parameter)	Runs in force mode.
j	Optional for moving objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for moving objects in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
exclude	Optional for moving objects in batches (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the object to be moved matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
include	Optional for moving objects in batches (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be moved does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is moved. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).
timeRange	Optional for moving objects in batches (additional parameter)	Indicates the time range matching pattern when moving objects. Only objects whose last modification time is within the configured time range are moved. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-***time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*****, all files whose last modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **mv_{succeed | failed | warning}_report_***time*_**TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Examples

- Take the Windows OS as an example. Run the **obsutil mv obs://bucket-test/key obs://bucket-test2** command to move a single object.

```
obsutil mv obs://bucket-test/key obs://bucket-test2

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

[=====================================================] 100.00% 6/s 0s
Move successfully, 19B, obs://bucket-test/key --> obs://bucket-test2/key
ext.txt
```

- Take the Windows OS as an example. Run the **obsutil mv obs://bucket-test/temp/ obs://bucket-test2 -f -r** command to move objects in batches.

```
obsutil mv obs://bucket-test/temp/ obs://bucket-test2 -f -r

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx
```

```
[============================================================] 100.00% 10/s 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:
9.71]
Task id is: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
```

4.8 Downloading an Object

Function

You can use this command to download an object or download objects in batches by object name prefix to your local PC.

NOTICE

- Do not change the source objects in the OBS bucket when downloading a single object or objects in batches. Otherwise, the download may fail or data may be inconsistent.
- If the storage class of the object to be copied is **cold**, you must restore the object to be downloaded first. Otherwise, the download fails.

Command Line Structure

- In Windows
  - Downloading a single object
    ```
    obsutil cp obs://bucket/key file_or_folder_url [-tempFileDir=xxx] [-
    dryRun] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-
    versionId=xxx] [-ps=auto] [-cpd=xxx][-fr] [-o=xxx] [-config=xxx]
    ```
  - Downloading objects in batches
    ```
    obsutil cp obs://bucket[/key] folder_url -r [-tempFileDir=xxx] [-dryRun]
    [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800]
    [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-
    mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```
- In Linux or macOS
  - Downloading a single object
    ```
    ./obsutil cp obs://bucket/key file_or_folder_url [-tempFileDir=xxx] [-
    dryRun] [-u] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-
    versionId=xxx] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Downloading objects in batches
    ```
    ./obsutil cp obs://bucket[/key] folder_url -r [-tempFileDir=xxx] [-
    dryRun] [-f] [-flat] [-u] [-vlength] [-vmd5] [-j=1] [-p=1] [-
    threshold=52428800] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-
    timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
file_or_folder_url	Mandatory for downloading an object	Local file/folder path

Parameter	Optional or Mandatory	Description
folder_url	Mandatory for downloading objects in batches	Local folder path
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Mandatory for downloading an object Optional for downloading objects in a batch	Indicates the name of the object to be downloaded, or the name prefix of the objects to be downloaded in batches. This parameter cannot be left blank when downloading an object. The saving and naming rules are as follows: ● If this parameter specifies a file or folder path that does not exist, the tool checks whether the value ends with a slash (/) or backslash (\). If yes, a folder is created based on the path, and the object is downloaded to this newly created directory. ● If this parameter specifies a file or folder path that does not exist and the value does not end with a slash (/) or backslash (\), the object is downloaded to your local PC with the value of **key** as the file name. ● If this parameter specifies an existing file, the object is downloaded to your local PC overwriting the existing file, with the value of **key** as the file name. ● If this parameter specifies an existing folder, the object is downloaded to the directory specified by **file_or_folder_url** with the object name as the file name. The saving rules when downloading objects in batches are as follows: ● If this parameter is left blank, all objects in the bucket are downloaded to the directory specified by **folder_url**. ● If this parameter is configured, objects whose name prefix is the configured value in the bucket are downloaded to the directory specified by **folder_url**. **NOTE** ● If this parameter is configured but the **flat** parameter is not configured when downloading objects in batches, the name of the downloaded file contains the name prefix of the parent object. If **flat** is configured, then the name of the downloaded file does not contain the name prefix of the parent object. ● For details about how to use this parameter, see **6.3 Download Examples**.

Parameter	Optional or Mandatory	Description
fr	Optional for downloading an object (additional parameter)	Generates an operation result list when downloading an object.
flat	Optional for downloading objects in batches (additional parameter)	The name prefix of the parent object is excluded when downloading objects in batches.
tempFileDir	Optional (additional parameter)	Indicates the directory for storing temporary files during multipart download. The default value is the value of **defaultTempFileDir** in the configuration file. **NOTE** ● If this parameter is left blank and the **defaultTempFileDir** parameter in the configuration file is also left blank, temporary files generated during multipart download are saved in the directory where to-be-downloaded files are located and end with the suffix of **.obs.temp**. ● Temporary files generated during multipart download are stored in this directory. Therefore, ensure that the user who executes obsutil has the write permission on the path. ● The available space of the partition where the path is located must be greater than the size of the objects to be downloaded.
dryRun	Optional (additional parameter)	Conducts a dry run.
u	Optional (additional parameter)	Indicates incremental download. If this parameter is set, each object can be downloaded only when it does not exist in the local path, its size is different from the namesake one in the local path, or it has the latest modification time.
vlength	Optional (additional parameter)	Checks whether the sizes of the local files are the same as those of the objects in the bucket after the download is complete.
vmd5	Optional (additional parameter)	Checks whether MD5 values of the local files are the same as those of the objects in the bucket after the download is complete. **NOTE** Objects in the bucket must contain metadata **x-obs-md5chksum**. Otherwise, MD5 verification will be skipped.

Parameter	Optional or Mandatory	Description
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart download tasks when downloading an object. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart download, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be downloaded is smaller than the threshold, download the object directly. If not, a multipart download is required. ● If you download an object directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
versionId	Optional for downloading an object (additional parameter)	Source object version ID that can be specified when downloading an object
ps	Optional (additional parameter)	Indicates the size of each part in a multipart download task, in bytes. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart download and saved to the **down** subfolder. After the download succeeds, its part record is deleted automatically. If the download fails or is suspended, the system attempts to resume the task according to its part record when you perform the download the next time.
r	Mandatory for downloading objects in batches (additional parameter)	Copies objects in batches based on a specified object name prefix.

Parameter	Optional or Mandatory	Description
f	Optional for downloading objects in batches (additional parameter)	Runs in force mode.
j	Optional for downloading objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for downloading objects in a batch. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
exclude	Optional for downloading objects in batches (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the object to be downloaded matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
include	Optional for downloading objects in batches (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be downloaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is downloaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).
timeRange	Optional for downloading objects in batches (additional parameter)	Indicates the time range matching pattern when downloading objects. Only objects whose last modification time is within the configured time range are downloaded. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*, all files whose last modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time*_**TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Examples

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/key d:\temp\test.txt** command to download a single object.

```
obsutil cp obs://bucket-test/key d:\temp\test.txt

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

[=========================================] 100.00% 4.86 KB/s 0s
Download successfully, 19B, obs://bucket-test/key --> d:\temp\test.txt
```

- Take the Windows OS as an example. Run the **obsutil cp obs://bucket-test/temp d:\ -f -r** command to download objects in batches.

```
obsutil cp obs://bucket-test/temp d:\ -f -r

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
```

```
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx

[=====================================================] 100.00% 155.59 KB/s 0s
Succeed count is:   6         Failed count is:    0
Metrics [max cost:153 ms, min cost:129 ms, average cost:92.00 ms, average tps:
17.86]
Task id is: 3066a4b0-4d21-4929-bb84-4829c32cbd0f
```

- For more examples, see **6.3 Download Examples**.

4.9 Generating the Download Link of an Object

Function

You can use this command to generate the download link of a specified object in a bucket or generate the download links of objects in a bucket in batches by object name prefix.

Command Line Structure

- In Windows
  - Generating the download link of a single object
    ```
    obsutil sign obs://bucket/key [-e=300] [-config=xxx]
    ```
  - Generating the download links of objects in batches by object name prefix
    ```
    obsutil sign obs://bucket[/key] -r [-e=300] [-timeRange=time1-time2] [-
    include=*.xxx] [-exclude=*.xxx] [-o=xxx] [-config=xxx]
    ```
- In Linux or macOS
  - Generating the download link of a single object
    ```
    ./obsutil sign obs://bucket/key [-e=300] [-config=xxx]
    ```
  - Generating the download links of objects in batches by object name prefix
    ```
    ./obsutil sign obs://bucket[/key] -r [-e=300] [-timeRange=time1-time2] [-
    include=*.xxx] [-exclude=*.xxx] [-o=xxx] [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Optional	Object name used for generating the download link of a single object, or object name prefix used for generating download links of objects in batches
e	Optional (additional parameter)	Validity period of the generated download links of objects, in seconds. Minimum value: 60s. Default value: 300s

Parameter	Optional or Mandatory	Description
r	Mandatory when generating download links of objects in batches (additional parameter)	Generates the download links of objects in batches by a specified object name prefix.
exclude	Optional when generating download links of objects in batches (additional parameter)	Indicates the matching patterns of objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the object to be downloaded matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
include	Optional when generating download links of objects in batches (additional parameter)	Indicates the matching patterns of objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be downloaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is downloaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).
timeRange	Optional (additional parameter)	Indicates the time range matching pattern when generating download links of objects. Only the download links of objects whose last modification time is within the configured time range are generated. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/). ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*****, all files whose last modification time is later than *time1* are matched.

Parameter	Optional or Mandatory	Description
o	Optional when generating download links of objects in batches (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **sign_{succeed | failed}_report_*time*_TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil sign obs://bucket-test/test.txt** command to generate the download link of a single object.

```
obsutil sign obs://bucket-test/test.txt

Download url of [obs://bucket-test/test.txt] is:
  http://your-endpoint/bucket-test/test.txt?
AccessKeyId=xxxx&Expires=1552548758&Signature=xxxx
```

4.10 Deleting an Object

Function

- You can use this command to delete a specified object.
- You can also use this command to delete objects in batches based on a specified object name prefix.

Command Line Structure

- In Windows
  - Deleting a single object
    ```
    obsutil rm obs://bucket/key [-f] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Deleting objects in batches
    ```
    obsutil rm obs://bucket/[key] -r [-j=1] [-f] [-v] [-o=xxx] [-config=xxx]
    ```
- In Linux or macOS
  - Deleting a single object
    ```
    ./obsutil rm obs://bucket/key [-f] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Deleting objects in batches

```
./obsutil rm obs://bucket/[key] -r [-j=1] [-f] [-v] [-o=xxx] [-
config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Mandatory for deleting a single object. Optional for deleting objects in batches.	Indicates the name of the object to be deleted, or the name prefix of the objects to be deleted in batches. **NOTE** If this parameter is left blank when deleting objects in batches, all objects in the bucket are deleted.
fr	Optional for deleting a single object (additional parameter)	Generates an operation result list when deleting an object.
f	Optional (additional parameter)	Runs in force mode.
versionId	Optional for deleting a single object (additional parameter)	Version ID of the object to be deleted.
r	Mandatory for deleting objects in batches (additional parameter)	Deletes objects in batches based on a specified object name prefix.
j	Optional for deleting objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for deleting objects in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
v	Optional for deleting objects in batches (additional parameter)	Deletes versions of an object and the delete markers in batches based on a specified object name prefix.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **rm_{succeed | failed}_report_***time*_**TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Examples

- Take the Windows OS as an example. Run the **obsutil rm obs://bucket-test/key -f** command to delete a single object named **key** in bucket **bucket-test**.

```
obsutil rm obs://bucket-test/key -f

Delete object [key] in the bucket [bucket-test] successfully!
```

- Take the Windows OS as an example. Run the **obsutil rm obs://bucket-test -r -f** command to delete all objects in bucket **bucket-test**.

```
obsutil rm obs://bucket-test -r -f

[=============================================] 100.00% 21s
Succeed count is:   1313      Failed count is:    0
Task id is: 95936984-f81a-441a-bba0-1fd8254d9241
```

4.11 Synchronously Uploading Incremental Objects

Function

This function synchronizes all content in the local source path to the specified target bucket on OBS, ensuring that the content is consistent between the local path and the target bucket. Incremental synchronization has the following meanings: 1) Increment: Compare the source file with the target object and upload only the source file that has changes. 2) Synchronization: After the command is executed, ensure that the local source path is a subset

of the target bucket specified by OBS. That is, any file in the local source path has its corresponding object in the target bucket on OBS.

NOTICE

- Do not change the local file or folder during synchronization. Otherwise, the synchronization may fail or data may be inconsistent.
- Each file can be synchronously uploaded only when it does not exist in the bucket, its size is different from the namesake one in the bucket, or it has the latest modification time.

Command Line Structure

- In Windows
  - Uploading a file synchronously

    ```
    obsutil sync file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-config=xxx]
    ```

  - Uploading a folder synchronously

    ```
    obsutil sync folder_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```

- In Linux or macOS
  - Uploading a file synchronously

    ```
    ./obsutil sync file_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-p=1] [-threshold=5248800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-config=xxx]
    ```

  - Uploading a folder synchronously

    ```
    ./obsutil sync folder_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
file_url	Mandatory for uploading a file synchronously	Local file path
folder_url	Mandatory for uploading a folder synchronously	Local folder path
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Optional	Indicates the object name or object name prefix specified when uploading a file synchronously, or the object name prefix specified when uploading a folder synchronously. The rules are as follows: ● If this parameter is left blank when synchronously uploading a file, the object is uploaded to the root directory of the bucket and the object name is the file name. If the value ends with a slash (/), the value is used as the object name prefix when the file is uploaded, and the object name is the value plus the file name. Otherwise, the file is uploaded with the value as the object name. ● If this parameter is left blank when synchronously uploading a folder, all objects in the root directory of the bucket are the same as the files in the local folder. If this parameter is configured, objects whose name prefix is the configured value are the same as the files in the local folder. **NOTE** ● If the value of this parameter does not end with a slash (/) when synchronously uploading a folder, the obsutil tool automatically adds a slash (/) at the end of the configured value as the object name prefix. ● For details about how to use this parameter, see **6.2 Synchronous Upload Examples**.
fr	Optional for synchronously uploading a file (additional parameter)	Generates an operation result list when synchronously uploading a file.
arcDir	Optional (additional parameter)	Path to which the synchronously uploaded files are archived
dryRun	Optional (additional parameter)	Conducts a dry run.

Parameter	Optional or Mandatory	Description
link	Optional (additional parameter)	Uploads the actual path of the symbolic-link file/folder **NOTICE** ● If this parameter is not specified and the file to be uploaded is a symbolic-link file whose target file does not exist, the exception message "The system cannot find the file specified" will be displayed in Windows OS, while the exception message "No such file or directory" will be displayed in macOS or Linux OS. ● Avoid the symbolic link loop of a folder, otherwise, the upload will exit due to panic. If you do not want the system to panic, set **panicForSymbolicLinkCircle** to **false** in the configuration file.
vlength	Optional (additional parameter)	After the synchronous upload is complete, check whether the sizes of the objects in the bucket are the same as those of the local files.
vmd5	Optional (additional parameter)	After the synchronous upload is complete, check whether the MD5 values of the objects in the bucket are the same as those of the local files. **NOTE** ● If the size of the file or folder to be uploaded is too large, using this parameter will degrade the overall performance due to MD5 calculation. ● After the MD5 value verification is successful, the parameter value is set to the object metadata **x-obs-md5chksum**, which is used for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart upload tasks when uploading a file. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart upload, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the file or folder to be uploaded is smaller than the threshold, upload it directly. Otherwise, a multipart upload is required. ● If you upload a file or folder directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.

Parameter	Optional or Mandatory	Description
acl	Optional (additional parameter)	Access control policies that can be specified when synchronously uploading files. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.
sc	Optional (additional parameter)	Indicates the storage classes of objects that can be specified when synchronously uploading files. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.
meta	Optional (additional parameter)	Indicates the customized metadata that can be specified when uploading files. The format is **key1:value1#key2:value2#key3:value3**. **NOTE** The preceding value indicates that the object in the bucket contains three groups of customized metadata after the file is uploaded: **key1:value1**, **key2:value2**, and **key3:value3**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart upload task, in bytes. The value ranges from 100 KB to 5 GB. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source file size.

Parameter	Optional or Mandatory	Description
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart upload and saved to the **upload** subfolder. After the upload succeeds, its part record is deleted automatically. If the upload fails or is suspended, the system attempts to resume the task according to its part record when you perform the upload the next time.
j	Optional for synchronously uploading a folder (additional parameter)	Indicates the maximum number of concurrent tasks for uploading a folder synchronously. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
exclude	Optional for synchronously uploading a folder (additional parameter)	Indicates the file matching patterns that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the file to be uploaded matches the value of this parameter, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder.

Parameter	Optional or Mandatory	Description
include	Optional for synchronously uploading a folder (additional parameter)	Indicates the file matching patterns that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be uploaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is uploaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern takes effect only for files in the folder.
timeRange	Optional for synchronously uploading a folder (additional parameter)	Indicates the time range matching pattern when synchronously uploading files. Only files whose last modification time is within the configured time range are uploaded. This pattern has a lower priority than the file matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured file matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*****, all files whose last modification time is later than *time1* are matched. **NOTICE** Time in the matching pattern is the UTC time.
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on folders.

Parameter	Optional or Mandatory	Description
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **sync_{succeed | failed | warning}_report_***time*_**TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Examples

- Take the Windows OS as an example. Run the **obsutil sync d:\temp\test.txt obs://bucket-test/key** command to synchronously upload a file.

```
obsutil sync d:\temp\test.txt obs://bucket-test/key

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx

[=================================================] 100.00% 1.68 MB/s 5s
Upload successfully, 8.46MB, d:\temp\test.txt --> obs://bucket-test/key
```

- Take the Windows OS as an example. Run the **obsutil sync d:\temp obs://bucket-test/temp** command to synchronously upload a folder.

```
obsutil sync d:\temp obs://bucket-test/temp

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx

[=================================================] 100.00% 2.02 KB/s 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71]
Task id is: 104786c8-27c2-48fc-bc6a-5886596fb0ed
```

- For more examples, see **6.2 Synchronous Upload Examples**.

4.12 Synchronously Copying Incremental Objects

Function

This function synchronizes all objects in the specified path in the source bucket with objects in the specified path in the destination bucket to keep data consistency. Incremental synchronization has the following meanings: 1) Increment: Compare the source object with the target object and copy only the source object that has changes. 2) Synchronization: After the command is executed, ensure that the specified path of the source bucket is a subset of the target bucket. That is, any object in the specified path of the source bucket has its corresponding object in the target bucket.

NOTICE

- Do not change the source objects in the OBS bucket during synchronization. Otherwise, the synchronization may fail or data may be inconsistent.
- If the storage class of the object to be copied is **cold**, you must restore the object to be copied first. Otherwise, the copy fails.
- To copy objects, you must have the read permission on the objects to be copied and the write permission on the destination bucket.
- If the client-side cross-region replication function is not enabled, ensure that the source bucket and destination bucket are in the same region.
- Each object can be synchronously copied only when it does not exist in the destination bucket, its size is different from the namesake one in the destination bucket, or it has the latest modification time.

Command Line Structure

- In Windows

```
obsutil sync obs://srcbucket[/key] obs://dstbucket[/dest] [-dryRun] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
```

- In Linux or macOS

```
./obsutil sync obs://srcbucket[/key] obs://dstbucket[/dest] [-dryRun] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
```

NOTE

The source path and destination path cannot be the same or nested when synchronously copying objects.

Parameter Description

Parameter	Optional or Mandatory	Description
srcbucket	Mandatory	Source bucket name

Parameter	Optional or Mandatory	Description
dstbucket	Mandatory	Destination bucket name
dest	Optional	Name prefix of destination objects. **NOTE** If the value of this parameter does not end with a slash (/), the obsutil tool automatically adds a slash (/) at the end of the configured value as the name prefix of destination objects.
key	Optional	Name prefix of source objects The rules are as follows: ● If this parameter is left blank, objects whose name prefix is the value of **dest** in the destination bucket are the same as all objects in the source bucket. ● If this parameter is configured, objects whose name prefix is the value of **dest** in the destination bucket are the same as objects whose name prefix is this configured value in the source bucket. **NOTE** ● If the value of this parameter does not end with a slash (/), the obsutil tool automatically adds a slash (/) at the end of the configured value as the name prefix of source objects. ● For details about how to use this parameter, see **6.6 Synchronous Copy Examples**.
dryRun	Optional (additional parameter)	Conducts a dry run.
crr	Optional (additional parameter)	Enables the client-side cross-region replication function. In this mode, data is directly copied to the destination bucket from the source bucket through data stream. The buckets can by any two OBS buckets. **NOTE** ● If this parameter is configured, ensure that the configuration of client-side cross-region replication is updated in the configuration file. For details, see **5.1 Updating a Configuration File**. ● The configurations of the source bucket and destination bucket are respectively **akCrr/skCrr/tokenCrr/endpointCrr** and **ak/sk/token/endpoint** in the configuration file. **NOTICE** After this function is enabled, both upload and download bandwidth are occupied.

Parameter	Optional or Mandatory	Description
vlength	Optional (additional parameter)	Verifies whether the object size in the destination bucket is the same as that in the source bucket after the copy task completes. **NOTE** This parameter must be used together with **crr**.
vmd5	Optional (additional parameter)	Verifies whether the MD5 value of the destination bucket is the same as that of the source bucket after the copy task completes. **NOTE** ● This parameter must be used together with **crr**. ● Objects in the source bucket must contain metadata **x-obs-md5chksum**. Otherwise, MD5 verification will be skipped. After the MD5 value verification is successful, the parameter value is set to the destination object metadata **x-obs-md5chksum**, which is used for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart copy tasks when copying an object. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart copy, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be copied is smaller than the threshold, copy the object directly. If not, a multipart copy is required. ● If you copy an object directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
acl	Optional (additional parameter)	Access control policies for destination objects that can be specified when copying objects. Possible values are: ● private ● public-read ● public-read-write ● bucket-owner-full-control **NOTE** The preceding four values indicate private read and write, public read, public read and write, and bucket owner full control.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Storage classes of the destination objects that can be specified when copying objects. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.
meta	Optional (additional parameter)	Metadata of destination objects that can be specified when copying objects. The format is *key1***:***value1***#***key2***:***value2***#***key3***:***value3*. **NOTE** The preceding value indicates that the destination objects in the bucket contain three groups of customized metadata after objects are copied: **key1:value1**, **key2:value2**, and **key3:value3**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart copy task, in bytes. The value ranges from 100 KB to 5 GB. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart copy and saved to the **copy** subfolder. After the copy succeeds, its part record is deleted automatically. If the copy fails or is suspended, the system attempts to resume the task according to its part record when you perform the copy the next time.

Parameter	Optional or Mandatory	Description
j	Optional for copying objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for copying objects synchronously. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
exclude	Optional for copying objects in batches (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the object to be copied matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
include	Optional for copying objects in batches (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be copied does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is copied. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).
timeRange	Optional (additional parameter)	Indicates the time range matching pattern when synchronously copying objects. Only objects whose last modification time is within the configured time range are copied. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*****, all files whose last modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **sync_{succeed | failed | warning}_report**_*time*_**TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Example

- Take the Windows OS as an example. Run the **obsutil sync obs://bucket-test/temp/ obs://bucket-test2 /temp/** command to synchronously copy objects.

```
obsutil sync obs://bucket-test/temp/ obs://bucket-test2/temp

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx


[============================================================] 100.00% 10/s 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:
9.71]
Task id is: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
```

- For more examples, see **6.6 Synchronous Copy Examples**.

4.13 Synchronously Downloading Incremental Objects

Function

This function synchronizes all content in the specified path of the source bucket to the target bucket on OBS, ensuring that the content is consistent between the specified path of the source bucket and the target bucket. Incremental synchronization has the following meanings: 1) Increment: Compare the source object with the target file and download only the source object that has changes. 2) Synchronization: After the command is executed, ensure that the specified path of the source bucket is a subset of the local target path. That is, any object in the specified path of the source bucket has its corresponding file in the local target path.

NOTICE

- Do not change the source objects in the OBS bucket during synchronization. Otherwise, the synchronization may fail or data may be inconsistent.
- If the storage class of the object to be copied is **cold**, you must restore the object to be downloaded first. Otherwise, the download fails.
- Each object can be synchronously downloaded only when it does not exist in the local path, its size is different from the namesake one in the local path, or it has the latest modification time.

Command Line Structure

- In Windows

```
obsutil sync obs://bucket[/key] folder_url [-tempFileDir=xxx] [-dryRun] [-
vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-
include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-
cpd=xxx] [-config=xxx]
```

- In Linux or macOS

```
./obsutil sync obs://bucket[/key] folder_url [-tempFileDir=xxx] [-dryRun] [-
vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-
include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-
cpd=xxx] [-config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
folder_url	Mandatory	Local folder path
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Optional	Indicates the name prefix of objects to be synchronously downloaded. The rules are as follows: ● If this parameter is left blank, all files in the folder specified by **folder_url** are the same as all objects in the bucket. ● If this parameter is configured, all files in the folder specified by **folder_url** are the same as the objects whose name prefix is the configured value in the bucket. **NOTE** ● If the value of this parameter does not end with a slash (/), the obsutil tool automatically adds a slash (/) at the end of the configured value as the object name prefix. ● For details about how to use this parameter, see **6.4 Synchronous Download Examples**.
tempFileDir	Optional (additional parameter)	Indicates the directory for storing temporary files during synchronous download. The default value is the value of **defaultTempFileDir** in the configuration file. **NOTE** ● Temporary files generated during multipart download are stored in this directory. Therefore, ensure that the user who executes obsutil has the write permission on the path. ● The available space of the partition where the path is located must be greater than the size of the objects to be downloaded.
dryRun	Optional (additional parameter)	Conducts a dry run.
vlength	Optional (additional parameter)	Checks whether the sizes of the local files are the same as those of the objects in the bucket after the download is complete.
vmd5	Optional (additional parameter)	Checks whether MD5 values of the local files are the same as those of the objects in the bucket after the download is complete. **NOTE** Objects in the bucket must contain metadata **x-obs-md5chksum**. Otherwise, MD5 verification will be skipped.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart download tasks when downloading an object. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart download, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be downloaded is smaller than the threshold, download the object directly. If not, a multipart download is required. ● If you download an object directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart download task, in bytes. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart download and saved to the **down** subfolder. After the download succeeds, its part record is deleted automatically. If the download fails or is suspended, the system attempts to resume the task according to its part record when you perform the download the next time.
j	Optional for downloading objects in batches (additional parameter)	Indicates the maximum number of concurrent tasks for downloading objects synchronously. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
exclude	Optional for downloading objects in batches (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the object to be downloaded matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
include	Optional for downloading objects in batches (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be downloaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is downloaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
timeRange	Optional (additional parameter)	Indicates the time range matching pattern when synchronously downloading objects. Only objects whose last modification time is within the configured time range are downloaded. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*, all files whose last modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **sync_{succeed | failed | warning}_report_***time***_TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.

Parameter	Optional or Mandatory	Description
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Example

- Take the Windows OS as an example. Run the **obsutil sync obs://bucket-test/temp d:\temp** command to download objects synchronously.

```
obsutil sync obs://bucket-test/temp d:\temp

Parallel:       3                   Jobs:           3
Threshold:      524288000           PartSize:       5242880
Exclude:                            Include:
VerifyLength:   false               VerifyMd5:      false
CheckpointDir: xxxx
OutputDir: xxxx

[=============================================================] 100.00% 155.59 KB/s 0s
Succeed count is:   6         Failed count is:    0
Metrics [max cost:153 ms, min cost:129 ms, average cost:92.00 ms, average tps:
17.86]
Task id is: 3066a4b0-4d21-4929-bb84-4829c32cbd0f
```

- For more examples, see **6.4 Synchronous Download Examples**.

4.14 Restoring Objects from OBS Cold

Function

You can use this command to restore a specified object whose storage class is **cold** or restore objects in batches by object name prefix.

NOTE

- Object content cannot be read during restoration.
- After an object is restored, the time it requires before the object can be downloaded depends on the OBS server.

Command Line Structure

- In Windows
  - Restoring an object

    ```
    obsutil restore obs://bucket/key [-d=1] [-t=xxx] [-versionId=xxx] [-fr]
    [-o=xxx] [-config=xxx]
    ```

  - Restoring objects in batches

    ```
    obsutil restore obs://bucket[/key] -r [-f] [-v] [-d=1] [-t=xxx] [-o=xxx]
    [-j=1] [-config=xxx]
    ```

- In Linux or macOS
  - Restoring an object
    ```
    ./obsutil restore obs://bucket/key [-d=1] [-t=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Restoring objects in batches
    ```
    ./obsutil restore obs://bucket[/key] -r [-f] [-v] [-d=1] [-t=xxx] [-o=xxx] [-j=1] [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
key	Mandatory for restoring a single object whose storage class is **cold** Optional for batch restoring objects whose storage class is **cold**	Indicates the name of the object to be restored or the name prefix of the objects to be restored in batches. **NOTE** If this parameter is left blank when batch restoring objects, all objects whose storage class is **cold** in the bucket are restored.
d	Optional (additional parameter)	Storage duration after objects whose storage class is **cold** are restored, in days. The value ranges from 1 to 30. The default value is 1.
t	Optional (additional parameter)	Options for restoring objects. Possible values are: ● standard ● expedited **NOTE** ● The preceding two values indicate standard restoration (3 - 5 hours) and quick restoration (1 - 5 minutes). ● If this parameter is not set, expedited restoration is used by default.
versionId	Optional for restoring a single object whose storage class is **cold** (additional parameter)	Version ID of the to-be-restored object whose storage class is **cold**
fr	Optional for restoring a single object whose storage class is **cold** (additional parameter)	Generates an operation result list when restoring a single object whose storage class is **cold**.

Parameter	Optional or Mandatory	Description
f	Optional for batch restoring objects whose storage class is **cold** (additional parameter)	Runs in force mode.
r	Mandatory for batch restoring objects whose storage class is **cold** (additional parameter)	Restores objects whose storage class is **cold** in batches by object name prefix.
v	Optional for batch restoring objects whose storage class is **cold** (additional parameter)	Restores versions of objects whose storage class is **cold** in batches by object name prefix.
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **restore_{succeed | failed}_report_***time*_**TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
j	Optional for batch restoring objects whose storage class is **cold** (additional parameter)	Maximum number of concurrent tasks for batch restoring objects whose storage class is **cold**. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Examples

- Take the Windows OS as an example. Run the **obsutil restore obs://bucket-test/key** command to restore a single object whose storage class is **cold**.

```
obsutil restore obs://bucket-test/key

Start to restore object [key] in the bucket [bucket-test] successfully!
```

- Take the Windows OS as an example. Run the **obsutil restore obs://bucket-test -r -f** command to restore objects whose storage class is **cold** in the bucket in batches.

```
obsutil restore obs://bucket-test -r -f

[=================================================] 100.00% 3s
Succeed count is:   12        Failed count is:    0
Metrics [max cost:264 ms, min cost:54 ms, average cost:119.33 ms, average tps:
19.70]
Task id is: 96f104ee-d0bf-40ff-95dd-31dec0d8f4f4
```

4.15 Resuming a Failed Upload Task

Function

You can use this command to resume a failed upload task based on the task ID.

Command Line Structure

- In Windows

```
obsutil cp -recover=xxx [-arcDir=xxx] [-dryRun] [-f] [-u] [-vlength] [-vmd5]
[-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-
timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
```

- In Linux or macOS

```
./obsutil cp -recover=xxx [-arcDir=xxx] [-dryRun] [-f] [-u] [-vlength] [-
vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-
timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
recover	Mandatory (additional parameter)	ID of the upload task to be resumed **NOTE** - You can obtain the task ID after an upload task is complete, or query it based on the file name of the operation result list, which is the 36 characters excluding the suffix **.txt** in the file name. - You can locate the upload task to be resumed in the directory where the result lists reside. For details about the directory of the result lists, see additional parameter **o**.
arcDir	Optional (additional parameter)	Path to which the uploaded files are archived

Parameter	Optional or Mandatory	Description
dryRun	Optional (additional parameter)	Conducts a dry run.
u	Optional (additional parameter)	Indicates incremental upload. If this parameter is set, each file can be uploaded only when it does not exist in the bucket, its size is different from the namesake one in the bucket, or it has the latest modification time.
vlength	Optional (additional parameter)	After the upload is complete, check whether the sizes of the objects in the bucket are the same as those of the local files.
vmd5	Optional (additional parameter)	After the upload completes, check whether the MD5 values of the objects in the bucket are the same as those of the local files. **NOTE** ● If the size of the file or folder to be uploaded is too large, using this parameter will degrade the overall performance due to MD5 calculation. ● After the MD5 value verification is successful, the parameter value is set to the object metadata **x-obs-md5chksum**, which is used for later MD5 verification during download or copy.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart upload tasks when uploading a file. The default value is the value of **defaultParallels** in the configuration file.
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart upload, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the file or folder to be uploaded is smaller than the threshold, upload it directly. Otherwise, a multipart upload is required. ● If you upload a file or folder directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
acl	Optional (additional parameter)	Access control policies that can be specified when uploading files. Possible values are: ● private ● public-read ● public-read-write **NOTE** The preceding three values indicate private read and write, public read, and public read and write.

Parameter	Optional or Mandatory	Description
sc	Optional (additional parameter)	Indicates the storage classes of objects that can be specified when uploading files. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.
meta	Optional (additional parameter)	Indicates the customized metadata that can be specified when uploading files. The format is **key1:value1#key2:value2#key3:value3**. **NOTE** The preceding value indicates that the object in the bucket contains three groups of customized metadata after the file is uploaded: **key1:value1**, **key2:value2**, and **key3:value3**.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart upload task, in bytes. The value ranges from 100 KB to 5 GB. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source file size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart upload and saved to the **upload** subfolder. After the upload succeeds, its part record is deleted automatically. If the upload fails or is suspended, the system attempts to resume the task according to its part record when you perform the upload the next time.
f	Optional (additional parameter)	Runs in force mode.

Parameter	Optional or Mandatory	Description
j	Optional (additional parameter)	Maximum number of concurrent tasks for uploading a folder. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
exclude	Optional (additional parameter)	Indicates the file matching patterns that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any characters, and question mark (?) represents only one character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the file to be uploaded matches the value of this parameter, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern applies only to files in a folder.
include	Optional (additional parameter)	Indicates the file matching patterns that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ****** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be uploaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is uploaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute file path (including the file name and file directory). ● The matching pattern applies only to files in a folder.

Parameter	Optional or Mandatory	Description
timeRange	Optional (additional parameter)	Indicates the time range matching pattern when uploading files. Only files whose last modification time is within the configured time range are uploaded. This pattern has a lower priority than the file matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured file matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*, all files whose last modification time is later than *time1* are matched. **NOTICE** Time in the matching pattern is the UTC time.
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Example

- Take the Windows OS as an example. Run the **obsutil cp -recover 104786c8-27c2-48fc-bc6a-5886596fb0ed -f** command to resume the failed upload task.

```
obsutil cp -recover 104786c8-27c2-48fc-bc6a-5886596fb0ed -f

Parallel:      3                    Jobs:          3
Threshold:     524288000            PartSize:      5242880
Exclude:                            Include:
VerifyLength:  false                VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx

[=============================================================] 100.00% 2.02 KB/s 0s
Succeed count is:   5         Failed count is:    0
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71]
Task id is: a628d6da-c562-4a1f-b687-4fa125de0dc3
```

4.16 Resuming a Failed Copy Task

Function

You can use this command to resume a failed copy task based on the task ID.

Command Line Structure

- In Windows

```
obsutil cp -recover=xxx [-dryRun] [-f] [-u] [-crr] [-vlength] [-vmd5] [-j=1]
[-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-meta=aaa:bbb#ccc:ddd] [-
ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-
o=xxx] [-cpd=xxx] [-config=xxx]
```

- In Linux or macOS

```
./obsutil cp -recover=xxx [-dryRun] [-f] [-u] [-crr] [-vlength] [-vmd5] [-
j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-sc=xxx] [-
meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-
timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
recover	Mandatory (additional parameter)	ID of the copy task to be resumed. **NOTE** ● You can obtain the task ID after a copy task is complete, or query it based on the file name of the operation result list, which is the 36 characters excluding the suffix **.txt** in the file name. ● You can locate the copy task to be resumed in the directory where the result lists reside. For details about the directory of the result lists, see additional parameter **o**.

Parameter	Optional or Mandatory	Description
dryRun	Optional (additional parameter)	Conducts a dry run.
crr	Optional (additional parameter)	Enables the client-side cross-region replication function. In this mode, data is directly copied to the destination bucket from the source bucket through data stream. The buckets can by any two OBS buckets. **NOTE** ● If this parameter is configured, ensure that the configuration of client-side cross-region replication is updated in the configuration file. For details, see **5.1 Updating a Configuration File**. ● The configurations of the source bucket and destination bucket are respectively **akCrr/skCrr/tokenCrr/endpointCrr** and **ak/sk/token/endpoint** in the configuration file. **NOTICE** After this function is enabled, both upload and download bandwidth are occupied.
vlength	Optional (additional parameter)	Verifies whether the object size in the destination bucket is the same as that in the source bucket after the copy task completes. **NOTE** This parameter must be used together with **crr**.
vmd5	Optional (additional parameter)	Verifies whether the MD5 value of the destination bucket is the same as that of the source bucket after the copy task completes. **NOTE** ● This parameter must be used together with **crr**. ● Objects in the source bucket must contain metadata **x-obs-md5chksum**. Otherwise, MD5 verification will be skipped. After the MD5 value verification is successful, the parameter value is set to the destination object metadata **x-obs-md5chksum**, which is used for later MD5 verification during download or copy.
u	Optional (additional parameter)	Indicates incremental copy. If this parameter is set, each object can be copied only when it does not exist in the destination bucket, its size is different from the namesake one in the destination bucket, or it has the latest modification time.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart copy tasks when copying an object. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart copy, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be copied is smaller than the threshold, copy the object directly. If not, a multipart copy is required. ● If you copy an object directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
acl	Optional (additional parameter)	Access control policies for destination objects that can be specified when copying objects. Possible values are: ● private ● public-read ● public-read-write **NOTE** The preceding three values indicate private read and write, public read, and public read and write.
sc	Optional (additional parameter)	Storage classes of the destination objects that can be specified when copying objects. Possible values are: ● **standard**: OBS Standard, which features low access latency and high throughput, and is applicable to storing frequently accessed data (multiple accesses per month averagely) or data that is smaller than 1 MB ● **warm**: OBS Warm. It is applicable to storing semi-frequently accessed (less than 12 times a year averagely) data that requires quick response. ● **cold**: OBS Cold. It is secure, durable, and inexpensive, and applicable to archiving rarely-accessed (once a year averagely) data.
meta	Optional (additional parameter)	Metadata of destination objects that can be specified when copying objects. The format is *key1***:***value1***#***key2***:***value2***#***key3***:***value3*. **NOTE** The preceding value indicates that the destination objects in the bucket contain three groups of customized metadata after objects are copied: **key1:value1**, **key2:value2**, and **key3:value3**.

Parameter	Optional or Mandatory	Description
ps	Optional (additional parameter)	Indicates the size of each part in a multipart copy task, in bytes. The value ranges from 100 KB to 5 GB. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart copy and saved to the **copy** subfolder. After the copy succeeds, its part record is deleted automatically. If the copy fails or is suspended, the system attempts to resume the task according to its part record when you perform the copy the next time.
f	Optional (additional parameter)	Runs in force mode.
j	Optional (additional parameter)	Indicates the maximum number of concurrent tasks for copying objects in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
exclude	Optional (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the object to be copied matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
include	Optional (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be copied does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is copied. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
timeRange	Optional (additional parameter)	Indicates the time range matching pattern when copying objects. Only objects whose last modification time is within the configured time range are copied. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*, all files whose last modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Example

- Take the Windows OS as an example. Run the **obsutil cp -recover=0476929d-9d23-4dc5-b2f8-0a0493f027c5 -f** command to copy objects in batches.

```
obsutil cp -recover=0476929d-9d23-4dc5-b2f8-0a0493f027c5 -f

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx


[============================================================] 100.00% 10/s 0s
Succeed count is:   1         Failed count is:    0
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:
9.71]
Task id is: f4c4f2b6-6e54-4dff-96b8-52e8c8c9a4b0
```

4.17 Resuming a Failed Download Task

Function

You can use this command to resume a failed download task based on the task ID.

Command Line Structure

- In Windows

```
obsutil cp -recover=xxx [-dryRun] [-tempFileDir=xxx] [-f] [-u] [-vlength] [-
vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-include=*.xxx] [-
exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-
config=xxx]
```

- In Linux or macOS

```
./obsutil cp -recover=xxx [-dryRun] [-tempFileDir=xxx] [-f] [-u] [-vlength] [-
vmd5] [-j=1] [-p=1] [-threshold=52428800] [-ps=auto] [-include=*.xxx] [-
exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-
config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
recover	Mandatory (additional parameter)	ID of the download task to be resumed **NOTE** ● You can obtain the task ID after a download task is complete, or query it based on the file name of the operation result list, which is the 36 characters excluding the suffix **.txt** in the file name. ● You can locate the download task to be resumed in the directory where the result lists reside. For details about the directory of the result lists, see additional parameter **o**.
tempFileDir	Optional (additional parameter)	Indicates the directory for storing temporary files during download. The default value is the value of **defaultTempFileDir** in the configuration file. **NOTE** ● Temporary files generated during multipart download are stored in this directory. Therefore, ensure that the user who executes obsutil has the write permission on the path. ● The available space of the partition where the path is located must be greater than the size of the objects to be downloaded.
dryRun	Optional (additional parameter)	Conducts a dry run.
u	Optional (additional parameter)	Indicates incremental download. If this parameter is set, each object can be downloaded only when it does not exist in the local path, its size is different from the namesake one in the local path, or it has the latest modification time.
vlength	Optional (additional parameter)	Checks whether the sizes of the local files are the same as those of the objects in the bucket after the download is complete.
vmd5	Optional (additional parameter)	Checks whether MD5 values of the local files are the same as those of the objects in the bucket after the download is complete. **NOTE** Objects in the bucket must contain metadata **x-obs-md5chksum**. Otherwise, MD5 verification will be skipped.
p	Optional (additional parameter)	Indicates the maximum number of concurrent multipart download tasks when downloading an object. The default value is the value of **defaultParallels** in the configuration file.

Parameter	Optional or Mandatory	Description
threshold	Optional (additional parameter)	Indicates the threshold for enabling multipart download, in bytes. The default value is the value of **defaultBigfileThreshold** in the configuration file. **NOTE** ● If the size of the object to be downloaded is smaller than the threshold, download the object directly. If not, a multipart download is required. ● If you download an object directly, no part record is generated, and resumable transmission is not supported. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.
ps	Optional (additional parameter)	Indicates the size of each part in a multipart download task, in bytes. The default value is the value of **defaultPartSize** in the configuration file. **NOTE** ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● The parameter can be set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.
cpd	Optional (additional parameter)	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** A part record is generated during a multipart download and saved to the **down** subfolder. After the download succeeds, its part record is deleted automatically. If the download fails or is suspended, the system attempts to resume the task according to its part record when you perform the download the next time.
f	Optional (additional parameter)	Runs in force mode.
j	Optional (additional parameter)	Indicates the maximum number of concurrent tasks for downloading objects in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.

Parameter	Optional or Mandatory	Description
exclude	Optional (additional parameter)	Indicates the matching patterns of source objects that are excluded, for example: ***.txt**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. For instance, **abc*.txt** indicates any file whose name starts with **abc** and ends with **.txt**. ● You can use ***** to represent ***** and **\?** to represent **?**. ● If the name of the object to be downloaded matches the value of this parameter, the object is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
include	Optional (additional parameter)	Indicates the matching patterns of source objects that are included, for example: ***.jpg**. **NOTE** ● The asterisk (*) represents any group of characters, and the question mark (?) represents any single character. ● You can use ***** to represent ***** and **\?** to represent **?**. ● Only after identifying that the name of the file to be downloaded does not match the value of **exclude**, the system checks whether the file name matches the value of this parameter. If yes, the file is downloaded. If not, the file is skipped. **NOTICE** ● You are advised to use quotation marks for the matching pattern to prevent special characters from being escaped by the OS and leading to unexpected results. Use single quotation marks for Linux or macOS and quotation marks for Windows. ● The matching pattern applies to the absolute path of an object, including the object name prefix and object name starting from the root directory. For example, if the path of an object in the bucket is **obs://bucket/src1/src2/test.txt**, then the absolute path of the object is **src1/src2/test.txt**. ● This matching pattern applies only to objects whose names do not end with a slash (/).

Parameter	Optional or Mandatory	Description
timeRange	Optional (additional parameter)	Indicates the time range matching pattern when downloading objects. Only objects whose last modification time is within the configured time range are downloaded. This pattern has a lower priority than the object matching patterns (**exclude**/**include**). That is, the time range matching pattern is executed after the configured object matching patterns. **NOTE** ● The matching time range is represented in *time1*-*time2*, where *time1* must be earlier than or the same as *time2*. The time format is *yyyyMMddHHmmss*. ● Automatic formatting is supported. For example, yyyyMMdd is equivalent to yyyyMMdd000000, and yyyyMM is equivalent to yyyyMM01000000. ● If this parameter is set to ***-*time2***, all files whose last modification time is earlier than *time2* are matched. If it is set to *time1*-*, all files whose last modification time is later than *time1* are matched. **NOTICE** ● Time in the matching pattern is the UTC time. ● This matching pattern applies only to objects whose names do not end with a slash (/).
mf	Optional (additional parameter)	Indicates that the name matching pattern (**include** or **exclude**) and the time matching pattern (**timeRange**) also take effect on objects whose names end with a slash (/).
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success, failure, and warning files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **cp_{succeed | failed | warning}_report_***time***_TaskId.txt** By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Example

- Take the Windows OS as an example. Run the **obsutil cp -recover=3066a4b0-4d21-4929-bb84-4829c32cbd0f d:\ -f -r** command to download objects in batches.

```
obsutil cp -recover=3066a4b0-4d21-4929-bb84-4829c32cbd0f -f -r

Parallel:      3                   Jobs:          3
Threshold:     524288000           PartSize:      5242880
Exclude:                           Include:
VerifyLength:  false               VerifyMd5:     false
CheckpointDir: xxxx
OutputDir: xxxx

[=====================================================] 100.00% 155.59 KB/s 0s
Succeed count is:   1          Failed count is:    0
Metrics [max cost:153 ms, min cost:129 ms, average cost:92.00 ms, average tps:
17.86]
Task id is: 19ad99ce-434e-41b2-9c8d-3af5b42eb65a
```

4.18 Listing Multipart Upload Tasks

Function

You can use this command to query multipart upload tasks in a bucket.

Command Line Structure

- In Windows

```
obsutil ls obs://bucket[/prefix] [-s] [-d] -m [-a] [-uploadIdMarker=xxx] [-
marker=xxx] [-limit=1] [-config=xxx]
```

- In Linux or macOS

```
./obsutil ls obs://bucket[/prefix] [-s] [-d] -m [-a] [-uploadIdMarker=xxx] [-
marker=xxx] [-limit=1] [-config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name
prefix	Optional	Object name prefix for listing multipart uploads **NOTE** If this parameter is left blank, all multipart upload tasks in the bucket are listed.

Parameter	Optional or Mandatory	Description
s	Optional (additional parameter)	Displays simplified query result. **NOTE** In the simplified format, the returned result contains only the object name and upload ID of the multipart upload.
d	Optional (additional parameter)	Lists only the multipart upload tasks and sub-directories in the current directory are listed, instead of recursively listing all the multipart upload tasks and sub-directories.
m	Mandatory (additional parameter)	Lists multipart upload tasks in the bucket.
a	Optional (additional parameter)	Lists the objects and the multipart upload tasks in the bucket.
marker	Optional (additional parameter)	Indicates the upload ID after which the multipart upload listing begins. All returned multipart upload tasks are listed in lexicographical order by object name involved in the tasks. **NOTE** For details about how to use this parameter, see **6.8 Examples of Listing Multipart Upload Tasks**.
uploadIdMarker	Optional (additional parameter)	Indicates the upload ID after which the multipart upload listing begins. This parameter must be used together with **marker**. All returned multipart upload tasks are listed in lexicographical order by object name and upload ID involved in the tasks.
limit	Optional (additional parameter)	Maximum number of objects that can be listed. If the value is less than or equal to **0**, all objects are listed. **NOTE** If there are a large number of multipart upload tasks in a bucket, you are advised to set this parameter to limit the number of multipart upload tasks each time. If not all tasks are listed, **marker** and **uploadIdMarker** of the next request will be returned in the result, which you can use to list the remaining tasks.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil ls obs://bucket-test -m -limit=10** command to query the multipart upload tasks in the bucket.

```
obsutil ls obs://bucket-test -m -limit=10
```

```
Listing multipart uploads.

Upload list:
Key                                                 Initiated
StorageClass        UploadId
obs://bucket-test/aaa                                   2018-11-27T03:49:07Z
standard            000001675348ED21860C3F61EF955BD3

obs://bucket-test/dir1/10GB.txt                         2018-11-07T06:58:09Z
standard            00000166ECF6CF7C860D1DBAF3F76013

obs://bucket-test/dir1/1GB.txt                          2018-11-07T06:58:09Z
standard            00000166ECF6CF6F860B7FBE95D01B03

obs://bucket-test/dir1/50GB.txt                         2018-11-07T06:58:09Z
standard            00000166ECF6CF86860D1DC2C8E8F66B

obs://bucket-test/dir1/5GB.txt                          2018-11-07T06:58:09Z
standard            00000166ECF6CF75860CDA7780CB52C3

obs://bucket-test/test11/20GB.txt                       2018-11-27T08:21:26Z
standard            0000016754423D24860CA8A4D06C2054

Folder number is: 0
Upload number is: 6
```

- For more examples, see **6.8 Examples of Listing Multipart Upload Tasks**.

4.19 Deleting a Multipart Upload Task

Function

- You can use this command to delete a multipart upload task in a specified bucket by using the multipart upload ID.
- You can also use this command to delete multipart upload tasks in batches based on a specified object name prefix.

Command Line Structure

- In Windows
  - Deleting a single multipart upload task
    ```
    obsutil abort obs://bucket/key -u=xxx [-f] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Deleting multipart upload tasks in batches
    ```
    obsutil abort obs://bucket[/key] -r [-f] [-o=xxx] [-j=1] [-config=xxx]
    ```
- In Linux or macOS
  - Deleting a single multipart upload task
    ```
    ./obsutil abort obs://bucket/key -u=xxx [-f] [-fr] [-o=xxx] [-config=xxx]
    ```
  - Deleting multipart upload tasks in batches
    ```
    ./obsutil abort obs://bucket[/key] -r [-f] [-o=xxx] [-j=1] [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
bucket	Mandatory	Bucket name

Parameter	Optional or Mandatory	Description
key	Mandatory for deleting a multipart upload task. Optional for deleting multipart upload tasks in batches.	Indicates the object name involved in a multipart upload task to be deleted, or the name prefix of the objects involved in multipart upload tasks to be deleted in batches. **NOTE** If this parameter is left blank when deleting multipart upload tasks in batches, all multipart upload tasks in the bucket are deleted.
u	Mandatory for deleting a single multipart upload task (additional parameter)	ID of the multipart upload task to be deleted **NOTE** You can obtain the value of this parameter from **4.18 Listing Multipart Upload Tasks**.
fr	Optional for deleting a single multipart upload task (additional parameter)	Generates an operation result list when deleting a multipart upload task.
f	Optional (additional parameter)	Runs in force mode.
r	Mandatory for deleting multipart upload tasks (additional parameter)	Deletes multipart upload tasks in batches based on a specified object name prefix.
j	Optional for deleting multipart upload tasks (additional parameter)	Indicates the maximum number of concurrent tasks for deleting multipart uploads in batches. The default value is the value of **defaultJobs** in the configuration file. **NOTE** The value is ensured to be greater than or equal to 1.
o	Optional (additional parameter)	Indicates the folder where operation result lists reside. After the command is executed, result lists (possibly including success and failure files) are generated in the folder. The default value is **.obsutil_output**, the subfolder in the home directory of the user who executes obsutil commands. **NOTE** ● The naming rule for result lists is as follows: **abort_{succeed | failed}_report_***time***_TaskId.txt** ● By default, the maximum size of a single result list is 30 MB and the maximum number of result lists that can be retained is 1024. You can set the maximum size and number by configuring **recordMaxLogSize** and **recordBackups** in the configuration file.

Parameter	Optional or Mandatory	Description
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Response

Refer to **Response** for uploading an object.

Running Examples

- Take the Windows OS as an example. Run the **obsutil abort obs://bucket-test/key -u=xxx -f** command to delete a single multipart upload task.

```
obsutil abort obs://bucket-test/key -u=xxx -f

Abort multipart upload [key] in the bucket [bucket-test] successfully!
```

- Take the Windows OS as an example. Run the **obsutil abort obs://bucket-test -r -f** command to delete all multipart upload tasks in the bucket in batches.

```
obsutil abort obs://bucket-test -r -f

Listing multipart uploads to abort...
Aborting progress:[============================================] 100.00% 3s
Succeed count is:   12        Failed count is:    0
Metrics [max cost:264 ms, min cost:54 ms, average cost:119.33 ms, average tps:
19.70]
Task id is: 0b34b1fa-b015-4313-a216-0fd5b4fffa1c
```

5 Auxiliary Commands

5.1 Updating a Configuration File

Function

You can use this command to update some configurations in the **.obsutilconfig** configuration file.

Command Line Structure

- In Windows
```
obsutil config -interactive [-crr] [-config=xxx]
```
- In Linux or macOS
```
./obsutil config -interactive [-crr] [-config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
interactive	Mandatory (additional parameter)	Updates configurations in interactive mode.
crr	Optional (additional parameter)	Updates the configurations related to client-side cross-region replication in the configuration file. **NOTE** If this parameter is set, fields in the configuration file corresponding to parameters **e**, **i**, **k**, and **t** are respectively changed to **endpointCrr**, **akCrr**, **skCrr**, and **tokenCrr**.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil config -interactive** command to set the access keys and endpoint of OBS.

```
obsutil config -interactive

Please input your ak:
xxxxxxxxxxxxxxxxxxxxxxxx
Please input your sk:
xxxxxxxxxxxxxxxxxxxxxxxx
Please input your endpoint:
xxxxxxxxxxxxxxxxxxxxxxxx
Please input your token:
xxxxxxxxxxxxxxxxxxxxxxxx
Config file url:
  C:\Users\tools\.obsutilconfig

Update config file successfully!
```

5.2 Deleting Part Records

Function

You can use this command to delete part records from a specified directory.

Command Line Structure

- In Windows

```
obsutil clear [checkpoint_dir] [-u] [-d] [-c] [-config=xxx]
```

- In Linux or macOS

```
./obsutil clear [checkpoint_dir] [-u] [-d] [-c] [-config=xxx]
```

Parameter Description

Parameter	Optional or Mandatory	Description
checkpoint_dir	Optional	Indicates the folder where the part records reside. The default value is **.obsutil_checkpoint**, the same subfolder where obsutil commands reside.
u	Optional (additional parameter)	Deletes the part records of all multipart upload tasks. **NOTE** At the same time, the system attempts to delete the multipart upload tasks in the part records.
d	Optional (additional parameter)	Deletes the part records of all multipart download tasks. **NOTE** At the same time, the system attempts to delete the fragments in the part records.
c	Optional (additional parameter)	Deletes the part records of all multipart copy tasks. **NOTE** At the same time, the system attempts to delete the multipart copy tasks in the part records.

Parameter	Optional or Mandatory	Description
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

NOTE

You must configure at least one among the **u**, **d** and **c** parameters.

Running Example

- Take the Windows OS as an example. Run the **obsutil clear -u** command to delete the part records of multipart upload tasks in the default directory.

```
obsutil clear -u

Clear checkpoint files for uploading in folder [xxxxx]
[=============================================================================] 100.00% 0s
Succeed files is:   1         Failed files is:    0
```

5.3 Viewing Command Help Information

Function

You can use this command to view the commands supported by obsutil or view the help information of a specific command.

Command Line Structure

- In Windows
  ```
  obsutil help [command]
  ```
- In Linux or macOS
  ```
  ./obsutil help [command]
  ```

Parameter Description

Parameter	Optional or Mandatory	Description
command	Optional	Currently, the help documents of the following commands are available: ● For **abort**, see **4.19 Deleting a Multipart Upload Task**. ● For **chattri**, see **4.4 Setting Object Properties**. ● For **cp**, see **4.2 Uploading an Object**, **4.6 Copying an Object**, and **4.8 Downloading an Object**. ● For **ls**, see **3.2 Listing Buckets**, **4.5 Listing Objects**, and **4.18 Listing Multipart Upload Tasks**. ● For **mb**, see **3.1 Creating a Bucket**. ● For **mkdir**, see **4.1 Creating a Folder**. ● For **mv**, see **4.7 Moving an Object**. ● For **restore**, see **4.14 Restoring Objects from OBS Cold**. ● For **rm**, see **3.5 Deleting a Bucket** and **4.10 Deleting an Object**. ● For **sign**, see **4.9 Generating the Download Link of an Object**. ● For **stat**, see **3.3 Querying Bucket Properties** and **4.3 Querying Object Properties**. ● For **sync**, see **4.11 Synchronously Uploading Incremental Objects**, **4.12 Synchronously Copying Incremental Objects**, and **4.13 Synchronously Downloading Incremental Objects**. ● For **archive**, see **5.5 Archiving Log Files**. ● For **clear**, see **5.2 Deleting Part Records**. ● For **config**, see **5.1 Updating a Configuration File**. ● For **help**, see **5.3 Viewing Command Help Information**. ● For **version**, see **5.4 Querying the Version Number**.

Running Example

- Take the Windows OS as an example. Run the **obsutil help mb** command to view the help information about the command for creating a bucket.

```
obsutil help mb
```

```
Summary:
create a bucket with the specified parameters

Syntax:
  obsutil mb obs://bucket [-acl=xxx] [-sc=xxx] [-location=xxx] [-config=xxx]

Options:
  -acl=xxx
    the ACL of the bucket, possible values are [private|public-read|public-read-
write]

  -sc=xxx
    the default storage class of the bucket, possible values are: [standard|warm|
cold]

  -location=xxx
    the region where the bucket is located

  -config=xxx
    the path to the custom config file when running this command
```

5.4 Querying the Version Number

Function

You can use this command to query the current version of obsutil.

Command Line Structure

- In Windows
  ```
  obsutil version
  ```
- In Linux or macOS
  ```
  ./obsutil version
  ```

Running Example

Take the Windows OS as an example.

```
obsutil version

obsutil version:5.1.9, obssdk version:2.2.12
operating system:windows, arch:amd64
```

5.5 Archiving Log Files

Function

You can use this command to archive log files to a local PC or to a specified bucket.

Command Line Structure

- In Windows
  - Archiving to a local PC
    ```
    obsutil archive [file_or_folder_url] [-config=xxx]
    ```
  - Archiving to a specified bucket
    ```
    obsutil archive obs://bucket[/key] [-config=xxx]
    ```

- In Linux or macOS
  - Archiving to a local PC
    ```
    obsutil archive [file_or_folder_url] [-config=xxx]
    ```
  - Archiving to a specified bucket
    ```
    obsutil archive obs://bucket[/key] [-config=xxx]
    ```

Parameter Description

Parameter	Optional or Mandatory	Description
file_or_folder_url	Optional	Indicates the path to which log files are archived. The rules are as follows: ● If this parameter is left blank, log files are archived to the same directory where obsutil commands reside with **obsutil_log.zip** as the archive file name. ● If this parameter specifies a file or folder path that does not exist, the tool checks whether the value ends with a slash (/) or backslash (\). If yes, a folder is created based on the path, and log files are archived to the newly created directory with **obsutil_log.zip** as the archive file name. ● If this parameter specifies a file or folder path that does not exist and the value does not end with a slash (/) or backslash (\), log files are archived to a local PC with the value as the archive file name. ● If this parameter specifies an existing .zip file, then log files are archived to a local PC overwriting the existing file, with the value as the archive file name. ● If this parameter specifies an existing folder, then log files are archived to the specified directory with **obsutil_log.zip** as the archive file name. **NOTE** All archive files are .zip files.
bucket	Mandatory for archiving log files to a specified bucket	Bucket name

Parameter	Optional or Mandatory	Description
key	Optional for archiving log files to a specified bucket	Indicates the object name or object name prefix when archiving log files to a specified bucket. The rules are as follows: ● If this parameter is left blank, log files are archived to the root directory of the bucket with **obsutil_log.zip** as the object name. ● If the value ends with a slash (/), the value is used as the object name prefix when archiving log files, and the object name is the value plus **obsutil_log.zip**. Otherwise, log files are archived with the value as the object name.
config	Optional (additional parameter)	User-defined configuration file for executing a command. For details about parameters that can be configured, see **A Parameter Description**.

Running Example

- Take the Windows OS as an example. Run the **obsutil archive** command to archive log files to the same directory where the tool is executed.

```
obsutil archive

[------------------------------------------------------------] 100.00% 15/15
35ms
Succeed to archive log files to [D:\obsutil\obsutil_log.zip]
```

6 Common Examples

6.1 Upload Examples

Assume that a local folder is in the following structure:

```
└── src1
    ├── src2
    │   ├── test1.txt
    │   └── test2.txt
    ├── src3
    └── test3.txt
```

Based on the preceding folder structure, different upload scenarios require different commands.

- To upload the **test3.txt** file in the local **src1** folder to the root directory of bucket **bucket-test**, the command is as follows:
  ```
  ./obsutil cp /src1/test3.txt  obs://bucket-test
  ```
  After the upload is successful, the following object is generated in the bucket:
  ```
  ./obs://bucket-test/test3.txt
  ```
- To upload the **test3.txt** file in the local **src1** folder to the root directory of bucket **bucket-test** and rename it to **aaa.txt**, the command is as follows:
  ```
  ./obsutil cp /src1/test3.txt  obs://bucket-test/aaa.txt
  ```
  After the upload is successful, the following object is generated in the bucket:
  ```
  ./obs://bucket-test/aaa.txt
  ```
- To upload the **test3.txt** file in the local **src1** folder to the **src** folder in bucket **bucket-test**, the command is as follows:
  ```
  ./obsutil cp /src1/test3.txt  obs://bucket-test/src/
  ```
  After the upload is successful, the following object is generated in the bucket:
  ```
  ./obs://bucket-test/src/test3.txt
  ```
- To recursively upload the entire local **src2** folder to the root directory of bucket **bucket-test** in force mode, the command is as follows:
  ```
  ./obsutil cp /src1/src2  obs://bucket-test -r -f
  ```
  After the upload is successful, the following objects are generated in the bucket:
  ```
  obs://bucket-test/src2/
  obs://bucket-test/src2/test1.txt
  obs://bucket-test/src2/test2.txt
  ```

- To recursively upload the entire local **src1** folder to the **src** folder in bucket **bucket-test** in force mode, the command is as follows:

  ```
  ./obsutil cp /src1  obs://bucket-test/src -r -f
  ```

  After the upload is successful, the following objects are generated in the bucket:

  ```
  obs://bucket-test/src/src1/
  obs://bucket-test/src/src1/src2/
  obs://bucket-test/src/src1/src2/test1.txt
  obs://bucket-test/src/src1/src2/test2.txt
  obs://bucket-test/src/src1/src3/
  obs://bucket-test/src/src1/test3.txt
  ```

- To recursively upload the all files and subfolders in the local **src1** folder to the **src** folder in bucket **bucket-test** in force mode, the command is as follows:

  ```
  ./obsutil cp /src1  obs://bucket-test/src -r -f -flat
  ```

  After the upload is successful, the following objects are generated in the bucket:

  ```
  obs://bucket-test/src/
  obs://bucket-test/src/src2/
  obs://bucket-test/src/src2/test1.txt
  obs://bucket-test/src/src2/test2.txt
  obs://bucket-test/src/src3/
  obs://bucket-test/src/test3.txt
  ```

NOTE

All the commands in the preceding examples use the Linux OS as the running environment.

6.2 Synchronous Upload Examples

Assume that a local folder is in the following structure:

```
└── src1
    ├── src2
        ├── test1.txt
        └── test2.txt
    ├── src3
    └── test3.txt
```

Assume that bucket **bucket-test** contains the following objects:

```
obs://bucket-test/src1/
obs://bucket-test/src1/src2/
obs://bucket-test/src1/src2/test1.txt
obs://bucket-test/src1/src3/
```

Based on the structure of the preceding local folder and objects in the bucket, different synchronous upload scenarios require different commands.

- To synchronize the **test3.txt** file in the local **src1** folder to the root directory of bucket **bucket-test**, the command is as follows:

  ```
  ./obsutil sync /src1/test3.txt  obs://bucket-test
  ```

  After the synchronization is successful, the **test3.txt** file is directly uploaded to the bucket because there is no **test3.txt** in bucket **bucket-test**. Then, objects in the bucket are as follows:

  ```
  obs://bucket-test/test3.txt
  obs://bucket-test/src1/
  obs://bucket-test/src1/src2/
  obs://bucket-test/src1/src2/test1.txt
  obs://bucket-test/src1/src3/
  ```

- To synchronize all files and subfolders in the local **src1** folder to the **src** folder in bucket **bucket-test**, the command is as follows:

  ```
  ./obsutil sync /src1  obs://bucket-test/src1
  ```

After the synchronization, the objects in the bucket are as follows:

```
obs://bucket-test/src1/
obs://bucket-test/src1/test3.txt
obs://bucket-test/src1/src2/
obs://bucket-test/src1/src2/test1.txt
obs://bucket-test/src1/src2/test2.txt
obs://bucket-test/src1/src3/
```

NOTE

All the commands in the preceding examples use the Linux OS as the running environment.

6.3 Download Examples

Assume that bucket **bucket-test** contains the following objects:

```
obs://bucket-test/test1.txt
obs://bucket-test/test2.txt
obs://bucket-test/test3.txt
obs://bucket-test/test4.txt
obs://bucket-test/test5.txt
obs://bucket-test/test6.txt
obs://bucket-test/src1/
obs://bucket-test/src1/test7.txt
obs://bucket-test/src2/
obs://bucket-test/src2/test8.txt
obs://bucket-test/src2/src3/
obs://bucket-test/src2/src3/test9.txt
```

Based on the structure of objects in the bucket, different download scenarios require different commands.

- To download the **test1.txt** file from bucket **bucket-test** to the local **src1** folder, the command is as follows:

  ```
  ./obsutil cp obs://bucket-test/test1.txt  /src1
  ```

  After the download is complete, the following file is generated on the local PC:

  ```
  └── src1
      └── test1.txt
  ```

- Run the following command to download the **test1.txt** file to your local PC. If there is no **test.txt** on the local PC, the **test1.txt** file is directly downloaded and you can rename it to **test.txt**. If **test.txt** already exists, **test1.txt** is downloaded and overwrites the original local **test.txt** file after renaming.

  ```
  ./obsutil cp obs://bucket-test/test1.txt  /test.txt
  ```

  After the download is complete, the following file generated on the local PC:

  ```
  └── test.txt
  ```

- To recursively download the entire **src2** folder from bucket **bucket-test** to the local **src1** folder in force mode, the command is as follows:

  ```
  ./obsutil cp obs://bucket-test/src2  /src1 -r -f
  ```

  After the download is complete, the following files are generated on the local PC:

  ```
  └── src1
      └── src2
          ├── src3
          │   └── test9.txt
          └── test8.txt
  ```

- To recursively download all files and subfolders in the **src2** folder from bucket **bucket-test** to the local **src1** folder in force mode, the command is as follows:

  ```
  ./obsutil cp obs://bucket-test/src2  /src1 -r -f -flat
  ```

  After the download is complete, the following files are generated on the local PC:

```
└── src1
    ├── src3
    │   └── test9.txt
    └── test8.txt
```

- To recursively download the all objects in bucket **bucket-test** to the local **src0** folder in force mode, the command is as follows:

```
./obsutil cp obs://bucket-test  /src0 -r -f
```

After the download is complete, the following files are generated on the local PC:

```
└── src0
    ├── test1.txt
    ├── test2.txt
    ├── test3.txt
    ├── test4.txt
    ├── test5.txt
    ├── test6.txt
    ├── src1
    │   └── test7.txt
    └── src2
        ├── src3
        │   └── test9.txt
        └── test8.txt
```

NOTE

All the commands in the preceding examples use the Linux OS as the running environment.

6.4 Synchronous Download Examples

Assume that bucket **bucket-test** contains the following objects:

```
obs://bucket-test/src1/
obs://bucket-test/src1/test3.txt
obs://bucket-test/src1/src2/
obs://bucket-test/src1/src2/test1.txt
obs://bucket-test/src1/src2/test2.txt
obs://bucket-test/src1/src3/
```

Assume that a local folder is in the following structure:

```
└── src1
    └── test3.txt
```

Based on the structure of the preceding local folder and objects in the bucket, different synchronous download scenarios require different commands.

- To synchronize all files and subfolders in the **src1** folder in bucket **bucket-test** to the local **src1** folder, the command is as follows:

```
./obsutil sync obs://bucket-test/src1  /src1
```

After the synchronization is successful, the following files are generated in the local **src1** folder:

```
└── src1
    ├── src2
    │   ├── test1.txt
    │   └── test2.txt
    ├── src3
    └── test3.txt
```

NOTE

All the commands in the preceding examples use the Windows OS as the running environment.

6.5 Copy Examples

Assume that bucket **bucket-src** contains the following objects:

```
obs://bucket-src/test1.txt
obs://bucket-src/test2.txt
obs://bucket-src/test3.txt
obs://bucket-src/test4.txt
obs://bucket-src/test5.txt
obs://bucket-src/test6.txt
obs://bucket-src/src1/
obs://bucket-src/src1/test7.txt
obs://bucket-src/src2/
obs://bucket-src/src2/test8.txt
obs://bucket-src/src2/src3/
obs://bucket-src/src2/src3/test9.txt
```

Based on the structure of objects in the bucket, different copy scenarios require different commands.

- To copy the **test1.txt** file from bucket **bucket-src** to bucket **bucket-dest**, the command is as follows:
  ```
  ./obsutil cp obs://bucket-src/test1.txt  obs://bucket-dest
  ```
  After the copy is complete, the following object is generated in bucket **bucket-dest**:
  ```
  obs://bucket-dest/test1.txt
  ```
- To copy the content of the **test1.txt** file in bucket **bucket-src** to the **text.txt** file in bucket **bucket-dest**, the command is as follows:
  ```
  ./obsutil cp obs://bucket-src/test1.txt  obs://bucket-dest/test.txt
  ```
  After the copy is complete, the following object is generated in bucket **bucket-dest**:
  ```
  obs://bucket-dest/test.txt
  ```
- To copy the **test1.txt** file in bucket **bucket-src** to the **text** folder in bucket **bucket-dest**, the command is as follows:
  ```
  ./obsutil cp obs://bucket-src/test1.txt  obs://bucket-dest/test/
  ```
  After the copy is complete, the following object is generated in bucket **bucket-dest**:
  ```
  obs://bucket-dest/test/test1.txt
  ```
- Run the following command to recursively copy the entire **src2** folder in bucket **bucket-src** to bucket **bucket-dest** in force mode:
  ```
  ./obsutil cp obs://bucket-src/src2  obs://bucket-dest -r -f
  ```
  After the copy is complete, the following objects are generated in bucket **bucket-dest**:
  ```
  obs://bucket-dest/src2/
  obs://bucket-dest/src2/test8.txt
  obs://bucket-dest/src2/src3/
  obs://bucket-dest/src2/src3/test9.txt
  ```
- To recursively copy all files and subfolders in the **src2** folder in bucket **bucket-src** to bucket **bucket-dest** in force mode, the command is as follows:
  ```
  ./obsutil cp obs://bucket-src/src2  obs://bucket-dest -r -f -flat
  ```
  After the copy is complete, the following objects are generated in bucket **bucket-dest**:
  ```
  obs://bucket-dest/test8.txt
  obs://bucket-dest/src3/
  obs://bucket-dest/src3/test9.txt
  ```

NOTE

All the commands in the preceding examples use the Linux OS as the running environment.

6.6 Synchronous Copy Examples

Assume that the source bucket **bucket-src** contains the following objects:

```
obs://bucket-src/src1/
obs://bucket-src/src1/test3.txt
obs://bucket-src/src1/src2/
obs://bucket-src/src1/src2/test1.txt
obs://bucket-src/src1/src2/test2.txt
obs://bucket-src/src1/src3/
```

Assume that the destination bucket **bucket-dest** contains the following objects:

```
obs://bucket-dest/src1/
obs://bucket-dest/src1/test3.txt
```

Based on the structure of objects in the bucket, different synchronous copy scenarios require different commands.

- To synchronize all files and subfolders in the **src1** folder in bucket **bucket-src** to the **src1** folder in bucket **bucket-dest**, the command is as follows:

  ```
  ./obsutil sync obs://bucket-src/src1  obs://bucket-dest/src1
  ```

  After the synchronous copy is complete, the objects in the destination bucket **bucket-dest** are as follows:

  ```
  obs://bucket-dest/src1/
  obs://bucket-dest/src1/test3.txt
  obs://bucket-dest/src1/src2/
  obs://bucket-dest/src1/src2/test1.txt
  obs://bucket-dest/src1/src2/test2.txt
  obs://bucket-dest/src1/src3/
  ```

NOTE

All the commands in the preceding examples use the Windows OS as the running environment.

6.7 List Examples

Assume that bucket **bucket-test** contains the following objects:

```
obs://bucket-test/test1.txt
obs://bucket-test/test2.txt
obs://bucket-test/test3.txt
obs://bucket-test/test4.txt
obs://bucket-test/test5.txt
obs://bucket-test/test6.txt
obs://bucket-test/src1/
obs://bucket-test/src1/test7.txt
obs://bucket-test/src2/
obs://bucket-test/src2/test8.txt
```

Based on the structure of objects in the bucket, different object listing scenarios require different commands.

- To list three objects in bucket **bucket-test**, the command is as follows:

  ```
  ./obsutil ls obs://bucket-test -limit=3
  ```

  The returned result is listed in lexicographical order by object name and version ID as follows:

  ```
  obs://bucket-test/test1.txt
  obs://bucket-test/test2.txt
  obs://bucket-test/test3.txt
  ```

- To list three objects following **test3.txt** in bucket **bucket-test**, the command is as follows:

  ```
  ./obsutil ls obs://bucket-test -limit=3 -marker=test3.txt
  ```

  The returned result is listed in lexicographical order by object name and version ID as follows:

  ```
  obs://bucket-test/test4.txt
  obs://bucket-test/test5.txt
  obs://bucket-test/test6.txt
  ```

- To list the files and subdirectories in the root directory of bucket **bucket-test** in non-recursive mode, that is, files in the subdirectories are not listed, the command is as follows:

  ```
  ./obsutil ls obs://bucket-test -d
  ```

  The returned result is listed in lexicographical order by object name and version ID as follows:

  ```
  obs://bucket-test/test1.txt
  obs://bucket-test/test2.txt
  obs://bucket-test/test3.txt
  obs://bucket-test/test4.txt
  obs://bucket-test/test5.txt
  obs://bucket-test/test6.txt
  obs://bucket-test/src1/
  obs://bucket-test/src2/
  ```

NOTE

All the commands in the preceding examples use the Linux OS as the running environment.

6.8 Examples of Listing Multipart Upload Tasks

Assume that bucket **bucket-test** contains the following multipart upload tasks:

```
obs://bucket-test/task1.txt uploadid1
obs://bucket-test/task1.txt uploadid2
obs://bucket-test/task2.txt uploadid3
obs://bucket-test/task3.txt uploadid4
obs://bucket-test/src1/
obs://bucket-test/src1/task4.txt uploadid5
obs://bucket-test/src2/
obs://bucket-test/src2/task5.txt uploadid6
```

- Run the following command to list three multipart upload tasks in bucket **bucket-test**:

  ```
  ./obsutil ls obs://bucket-test -m -limit=3
  ```

  The returned result is listed in lexicographical order by object name as follows:

  ```
  obs://bucket-test/task1.txt uploadid1
  obs://bucket-test/task1.txt uploadid2
  obs://bucket-test/task2.txt uploadid3
  ```

- To list the rest multipart upload tasks following **uploadid1**, the command is as follows:

  ```
  obsutil ls obs://bucket-test -m -limit=3 -marker=task1.txt -
  uploadIdMarker=uploadid1
  ```

  The returned result is listed in lexicographical order by object name and upload ID as follows:

  ```
  obs://bucket-test/task1.txt uploadid2
  obs://bucket-test/task2.txt uploadid3
  obs://bucket-test/task3.txt uploadid4
  ```

NOTE

All the commands in the preceding examples use the Linux OS as the running environment.

7 Fault Locating

7.1 Overview

obsutil provides multiple methods for users to locate and analyze faults. **Table 7-1** details the methods. Generally, you need to combine these methods for a precise fault locating.

Table 7-1 Fault locating methods

Method	Description
7.2 Log Files	obsutil log files include tool logs and SDK logs. The tool logs record the success information and exceptions generated during obsutil running. The SDK logs record the success information and exceptions generated during requesting for OBS.
7.3 Result Lists	Result lists are generated after batch tasks complete and may include success, failure, and warning files.
7.4 Return Codes	obsutil yields different return codes based on different execution results. You can analyze and troubleshoot faults according to these return codes.

7.2 Log Files

Configuring Log Files

obsutil log files include tool logs and SDK logs. You can add the following parameters to the **.obsconfigutil** file to enable the two logging functions.

- Tool logging (records the log information generated during obsutil running): configure **utilLogPath**, **utilLogBackups**, **utilLogLevel**, and **utilMaxLogSize**.
- SDK logging (records the log information generated when using obsutil to call OBS server-side APIs): configure **sdkLogPath**, **sdkLogBackups**, **sdkLogLevel**, and **sdkMaxLogSize**.

NOTE

- For details about the parameter description, see **A Parameter Description**.
- **utilLogPath** and **sdkLogPath** indicate the absolute paths of the log files, not the folders that store the log files.
- If **utilLogPath** and **sdkLogPath** are not specified, tool logging and SDK logging are not enabled, and therefore no log file is generated during obsutil running.
- Log files that are rolled over are named as follows: *filename***.log.***number*

NOTICE

If multiple obsutil processes are running at the same time, log files may fail to be written concurrently or may be lost. In this case, add parameter **-config** when running commands to configure an independent configuration file for each process. Make sure that **utilLogPath** and **sdkLogPath** are set to different paths for each process.

Collecting Log Files

You can collect logs in either of the following methods:

Method 1: Use auxiliary commands by referring to **5.5 Archiving Log Files**.

Method 2: Locate the paths specified by **utilLogPath** and **sdkLogPath** in the configuration file, and then search for the log files in the corresponding paths in the local file system.

7.3 Result Lists

Configuring Result Lists

Result lists are generated when batch tasks complete. By default, they are saved to the subfolder **.obsutil_output** in the home directory of the user who executes obsutil commands. You can specify another folder to save them by setting the additional parameter **-o** when executing a command.

Viewing Result Lists

Result lists are classified into success, failure, and warning lists. The naming rule is as follows: *Operation* **_{succeed | failed | warning}_report_** *Time* **_TaskId.txt**. For example, after a folder is successfully uploaded, the result list is named as follows: **cp_succeed_report_20190417021908_fbbc83e3-98ac-4d19-b23a-64023b1e0c34.txt**.

NOTE

- If the number of successes, failures, or warnings is zero, the corresponding result list is not generated.
- The task ID of a result list is unique for each operation.
- The maximum size of a result list is 30 MB and the maximum number of lists that can be retained is 1024.

7.4 Return Codes

If obsutil is invoked by processes, the command output cannot be viewed in real time. obsutil generates different return codes based on different execution results. Table 7-2 describes the return codes. You can use either the following methods to obtain the return code of the latest execution result and then analyze and rectify the fault based on it:

- In the macOS or Linux OS, run the following command to obtain the return code of the latest execution result:
  ```
  echo $?
  ```
- In the Windows OS, run the following command to obtain the return code of the latest execution result:
  ```
  echo %errorlevel%
  ```

Table 7-2 Return codes

Return Code	Meaning	Example Scenario
0	Execution succeeded.	An object is successfully uploaded.
1	The file does not exist.	The entered file path does not exist for uploading a file by running the **cp** command.
2	The task does not exist.	The specified task ID does not exist for resuming a failed upload task by running the **cp** command.
3	Parameter error	● At least one entered additional parameters is not supported for uploading a file by running the **cp** command. ● The entered value of **cloud_url** is invalid for downloading a file by running the **cp** command. **NOTE** **cloud_url** indicates the bucket path or object path. Set **cloud_url** in the format of **obs://***bucketname* when downloading all objects in a bucket. Set **cloud_url** in the format of **obs://***bucketname/key* when downloading a specified object in a bucket.
4	Bucket status error	The specified destination bucket does not exist for uploading a folder by running the **cp** command.
5	Initialization error during command execution	● An error occurs when loading the configuration file. ● Parameter **-o** is configured when running the **cp** command to upload a folder, but the folder specified by **-o** for saving the result lists fails to be created.
6	Execution error.	When you run the **ls** command to query the bucket list, the query fails because the network times out.

Return Code	Meaning	Example Scenario
7	The operation is not supported.	The version of the bucket is not 3.0 when you attempt to change the properties of an object in it by running the **chattri** command.
8	A batch task succeeded partially.	Some objects fail to be downloaded during a batch download by running the **cp** command.
9	Interruption error	Users press **Ctrl** + **C** to interrupt the command execution.
-1	Unknown error	-

8 Best Practices

8.1 Using the obsutil help Command to Search for Functions

obsutil provides **help** commands for viewing the help documents of each command. To query the help document of the bucket creation command, perform the following steps:

Step 1 Run the **obsutil help** command to query the list of all supported commands.

Step 2 Find the abbreviation of the command to be viewed based on the document description in the command list. For example, the abbreviation of the command for creating a bucket is **mb**.

Step 3 Run the **obsutil help mb** command to view the usage and detailed functions of the **mb** command, illustrated as follows:

```
Summary:
create a bucket with the specified parameters

Syntax:
  obsutil mb obs://bucket [-acl=xxx] [-sc=xxx] [-location=xxx] [-config=xxx]

Options:
  -acl=xxx
    the ACL of the bucket, possible values are [private|public-read|public-read-
write]

  -sc=xxx
    the default storage class of the bucket, possible values are: [standard|warm|
cold]

  -location=xxx
    the region where the bucket is located

  -config=xxx
    the path to the custom config file when running this command
```

Step 4 Run the **obsutil mb obs://bucket-test -location xxx** command to create a bucket named **bucket-test** in the *xxx* region.

----End

NOTE

- For more information about the **help** command, see **5.3 Viewing Command Help Information**.
- You can set the **helpLanguage** parameter in the configuration file to configure the language type of the **help** command. For example, **helpLanguage=Chinese** indicates that the language type of the help command is Chinese.
- The supported languages are Chinese and English. The default language is English.

8.2 Configuring Scheduled Tasks Using the Crontab Command

Scenario

Go to the **/root** directory at 21:30 every day and upload the **/src/src1** folder to bucket **obs://bucket-test** in the incremental mode.

Prerequisites

You have properly enabled the scheduled crond service in the Linux OS.

NOTE

Run the **service crond status** command to check whether the service is enabled.

Procedure

Step 1 Run the **crontab -e** command to open the configuration file for setting a scheduled task.

Step 2 Enter the Insert mode to edit the configuration file.

```
30 21 * * * cd /root && nohup ./obsutil cp /src/src1 obs://bucket-test -r -f -u
&>obsutil_crond.log &
```

NOTE

Assume that the obsutil tool is in the **/root** directory. The preceding configuration is described as follows: Go to the **/root** directory at 21:30 every day, upload the **/src/src1** folder to bucket **obs://bucket-test** in incremental mode, and redirect the command output to the **obsutil_crond.log** file in the **/root** directory.

Step 3 Press **Esc** to exit the Insert mode. Then input **:wq** and press **Enter** to save the configuration and exit.

Step 4 Run the **crontab -l** command to check whether the scheduled task is configured successfully.

----End

FAQs

1. How do I determine whether a scheduled task is being executed?
   - Run the **tail /var/log/cron** command to view the latest scheduled task execution records.
   - Run the **ps -ef | grep obsutil** command to check whether obsutil is being executed.
2. How do I forcibly stop an ongoing scheduled task?
   a. Run the **ps -ef | grep obsutil** command to check the process of obsutil.

b. Run the **kill -9** *PID* command to forcibly stop the process, where *PID* indicates the queried process ID.

8.3 Setting obsutil Commands to Built-in Commands

Scenario

Because obsutil is external software, you need to access the directory where obsutil resides before running obsutil commands. In this way, the usability of the tool is poor.

An OS provides built-in commands so that directories which support running of the commands are loaded to the memory when the system is started. In this way, you can run commands in any directory, which improves the tool's usability.

This section introduces how to set obsutil commands to built-in commands in different OSs.

Setting obsutil Commands to Built-in Commands in Windows

Step 1 In the CLI, run the **echo %PATH%** command to query all the paths configured in the current system. Then select one as the operation path.

Step 2 Run the **mklink** *PATH*/**obsutil.exe** *OBSUTIL_PATH* command to set obsutil commands to built-in commands of the system.

NOTE

PATH indicates the operation path selected in step 1. *OBSUTIL_PATH* indicates the absolute path of **obsutil.exe**.

Step 3 Check whether the configuration is successful: Run the **obsutil help** command in the CLI. If the help information is displayed, the configuration is successful.

----End

Setting obsutil Commands to Built-in Commands in Linux or macOS

Step 1 Run the following command to create a directory for the obsutil tool:

```
mkdir /obsutil
```

NOTE

- Skip this step if the directory already exists.
- You must run the command as user **root**.

Step 2 Run the following command to grant the **755** permission for the tool's directory:

```
chmod 755 /obsutil
```

NOTE

- Skip this step if the permission for the directory is **drwxr-xr-x**.
- You must run the command as user **root**.

Step 3 Copy the obsutil tool to the directory created in step 1 and change its permission to **711**. Assume that the original path of the tool is **/home/test/obsutil**. Run the following command:

```
cp /home/test/obsutil /obsutil
chmod 711 /obsutil/obsutil
```

Step 4 Run the **vi /etc/profile** command, type **I** to enter the Insert mode to edit the file. Add **export PATH=$PATH:/obsutil** at the end of the file. Then press **ESC** to exit the editing mode, and then type **:wq!** and press **Enter** to save the file and exit.

NOTE

Skip this step if the new line already exists in the **/etc/profile** file.

Step 5 Run the **echo $PATH** command to query the current environment variables. If **:/obsutil** in included in the query result, indicating that the **/obsutil** environment variable already exists, go to the next step. Otherwise, run the **source /etc/profile** command.

Step 6 Check whether the configuration is successful: Run the **obsutil help** command in any directory. If the help information is displayed, the configuration is successful.

----End

FAQs

1. How do I locate the obsutil configuration file after setting obsutil commands to built-in commands?

   The **.obsutilconfig** file in the same directory where obsutil commands reside is the configuration file of the obsutil tool. You can also run the **obsutil config** command to obtain the configuration file path. An example is provided as follows:

   ```
   obsutil config
   Config file url:
     D:\tools\.obsutilconfig
   ```

2. How do I delete obsutil commands after setting them as built-in commands?
   - In Windows:
     i. Run the **where obsutil** command to locate the path of obsutil commands.
     ```
     where obsutil
     E:\tools\bin\obsutil.exe
     ```
     ii. Run the **del** *PATH* command to delete obsutil commands.
     ```
     del E:\tools\bin\obsutil.exe
     ```
     NOTE

     Replace *PATH* with the path of obsutil commands. **E:\tools\bin\obsutil.exe** is used in the preceding example.
   - In Linux or macOS:
     i. Run the **which obsutil** command to locate the path of obsutil commands.
     ```
     which obsutil
     /obsutil/obsutil
     ```
     ii. Run the **rm -rf** *PATH* command to delete obsutil commands.
     ```
     rm -rf /obsutil/obsutil
     ```
     NOTE

     Replace *PATH* with the path of obsutil commands. **/obsutil/obsutil** is used in the preceding example.

     iii. Restore the system environment variable: Delete the path of obsutil that is set in the **/etc/profile** file.

     NOTE

     If the **/etc/profile** file contains line **export PATH=$PATH:/obsutil**, delete the line. Or if the file contains line **export PATH=$PATH:/test/bin:/obsutil:/test1**, delete **:/obsutil** from the line.

3. What should I do if the execution of built-in obsutil commands fails in Linux or macOS?
   - If the message **Permission denied** is displayed after executing **obsutil help**, run the **chmod 755** *OBSUTIL_PATH* command (replace *OBSUTIL_PATH* with the path of obsutil) to add an execute permission for the obsutil tool.
   - If the message **command not found** is displayed, log in again.
   - If the message **Cannot create parent folder for xx/.obsutilconfig, xx Permission denied** is displayed, check whether the home directory of the user exists.

   NOTICE

   In the Ubuntu OS, if you run the **useradd** command to add a user, the home directory of the user is not created by default. You need to create it manually. Therefore, you are advised to run the **adduser** command to add a user.

4. What can I do if no log file is generated after running built-in obsutil commands in Linux or macOS?

   If you have properly configured **sdkLogPath** and **utilLogPath** in the configuration file, but still no log file is generated after command execution, then check whether the user who runs the command has the read and write permissions on **sdkLogPath** and **utilLogPath**.

8.4 Fine-Tuning obsutil Performance

By default, obsutil uploads, downloads, and copies files or objects whose size is greater than 50 MB in multiple parts. Table 8-1 details related parameters in the **.obsutilconfig** file.

Table 8-1 Multipart-related parameters

Parameter	Description
defaultBigfileThreshold	Indicates the threshold for triggering multipart tasks, in bytes. If the size of a file to be uploaded, downloaded, or copied is greater than the threshold, the file is uploaded, downloaded, or copied in multiple parts. The default value is 50 MB.
defaultPartSize	Size of each part, in bytes. The default value is **auto**. **NOTE** • For multipart upload and copy, the value ranges from 100 KB to 5 GB. • For multipart download, the value is unrestricted.
defaultParallels	Maximum number of concurrent tasks in the multipart mode. The default value is 5.

Generally, multipart tasks not only speed up transmission but also allow you to resume failed tasks. By default, the part size of a multipart task can be automatically adjusted by the obsutil in the **auto** mode. In practice, however, to further improve the upload and download performance, you can adjust the part size according to the file size and the network conditions, to obtain the maximum transmission efficiency and ensure the successful completion of a transmission task.

Adjust the number of concurrent tasks in the multipart mode according to the following formula:

defaultParallels = Min(Number of CPUs x 2, Object size/defaultPartSize x 1.5)

In the upload, download, and copy commands, parameters **-p** and **-ps** are used to modify the number of concurrent tasks in the multipart mode and part size respectively, and then deliver the multipart task based on the parameter values configured in the command. The default values in the configuration file are used if you do not set them in a command.

Adjust the number of concurrent tasks in the multipart mode according to the following formula:

p = Min(Number of CPUs x 2, Object size/ps x 1.5)

NOTE

- Resources of a running host are limited. Therefore, if the number of concurrent tasks in the multipart mode is set too large, the performance of obsutil upload, download, or copy may deteriorate due to resource switchover and preemption between threads. In this case, you need to adjust the values of **defaultParallels** (**-p**) and **defaultPartSize** (**-ps**) based on the actual file size and network status. To perform a pressure test, lower the two values at first, and then gradually increase them to determine the optimal values.
- If the values of **defaultParallels** (**-p**) and **defaultPartSize** (**-ps**) are too large, an EOF error may occur due to network instability. In this case, set the two parameters to smaller values.
- If a batch operation is performed, the destination object size can be set to the average size of the objects to be operated.

8.5 Using obsutil for Resumable Data Transfer

obsutil supports resumable data transfer (upload, download, and copy) for large files by using the multipart algorithms for upload, download, and copy. You can set the threshold size for starting a multipart upload, download, or copy task based on your actual requirements to resume the upload, download, or copy task if the task fails or is interrupted. You can specify the threshold size for starting a multipart task in either of the following ways:

1. Specify the **defaultBigfileThreshold** parameter in the configuration file.
2. When running commands for upload, download, copy, incremental synchronization upload, incremental synchronization download, or incremental synchronization copy, you can specify the **threshold** parameter at the command level.

NOTE

- Priority: Command level parameter **threshold** has higher priority than the **defaultBigfileThreshold** in the configuration file.
- The threshold size of a multipart task applies to single files or objects. When the size of a file or object is greater than the threshold value, the multipart algorithm is applied to the file or object.
- The multipart algorithm and resumable data transfer are forcibly bound together. That is, once the multipart algorithm is used, the resumable data transfer is enabled for the task.

8.6 Using obsutil to Upload a Symbolic Link

obsutil supports the upload of the real path to which the symbolic link points when a file or folder is uploaded. You can specify the command-level parameter **link** to implement this function when running commands for upload or incremental synchronization upload.

NOTE

- obsutil can identify symbolic links pointing to folders. If a symbolic link points to a folder, obsutil recursively scans the contents in the folder.
- Avoid the symbolic link loop of a folder, otherwise, the upload will exit due to panic. If you do not want the system to panic, set **panicForSymbolicLinkCircle** to **false** in the configuration file.
- The symbolic link and the shortcut on the Windows OS are two different types. obsutil cannot identify the shortcut on the Windows OS.

8.7 Configuring an HTTP Proxy for obsutil

You can configure an HTTP proxy in either of the following ways:

Method 1: Set the **proxyUrl** parameter in the **.obsutilconfig** file. Example: **proxyUrl=http://username:password@your-proxy:8080**;

Method 2: Use the system environment variable **HTTPS_PROXY** or **HTTP_PROXY**. Example: **HTTPS_PROXY=http://username:password@your-proxy:8080**.

NOTE

- HTTP proxy format: **http://[*Username*:*Password*@]*Proxy server address*:*Port number***. The *Username* and *Password* are optional.
- The **proxyUrl** parameter and system environment variables are in the following priority order: **proxyUrl** > **HTTPS_PROXY** > **HTTP_PROXY**.
- The user name and password cannot contain colons (:) and at signs (@), which will result in parsing errors.

A Parameter Description

You can use the **.obsutilconfig** file to configure the parameters of obsutil. The following table lists detailed information about the parameters.

Table A-1 obsutil parameters

Parameter	Optional or Mandatory	Description	Recommended Value
endpoint	Mandatory	Endpoint for accessing OBS, which can contain the protocol type, domain name, and port number (optional). Example: **https://your-endpoint:80** **NOTE** ● You can click **here** to view the endpoints and regions enabled for OBS. ● If the configured endpoint does not contain any protocol, the HTTPS protocol is used by default.	N/A
ak	Mandatory	Access key ID **NOTE** ● After you run obsutil for the first time, the tool encrypts the AK to ensure the key security.	N/A
sk	Mandatory	Secret access key **NOTE** ● After you run obsutil for the first time, the tool encrypts the SK to ensure the key security.	N/A

Parameter	Optional or Mandatory	Description	Recommended Value
endpointCrr	Optional	Endpoint for accessing OBS in the region where the source bucket resides when the client-side cross-region replication function is enabled, which can contain the protocol type, domain name, and port number. Example: **http://your-endpoint:80** **NOTE** ● You can click **here** to view the endpoints and regions enabled for OBS. ● If the configured endpoint does not contain any protocol, the HTTPS protocol is used by default.	N/A
akCrr	Optional	AK for the source bucket when the client-side cross-region replication function is enabled	N/A
skCrr	Optional	SK for the source bucket when the client-side cross-region replication function is enabled	N/A
token	Optional	Security token. If this parameter is left blank, the security token is not set.	N/A
connectTimeout	Optional	Timeout interval for establishing an HTTP/HTTPS connection, in seconds. The default value is 30.	The recommended value ranges from 5 to 120.
socketTimeout	Optional	Timeout interval for reading and writing data, in seconds. The default value is 310.	The recommended value ranges from 5 to 600.
maxRetryCount	Optional	Maximum number of retry attempts. The default value is 3. **NOTE** When an OBS request completes but HTTP status code 408 or 5XX is returned, or when a timeout error occurs to an OBS request, the request is retried.	The recommended value ranges from 0 to 5.
maxConnections	Optional	Maximum number of HTTP connections that can be accessed. The default value is 1000.	N/A

Parameter	Optional or Mandatory	Description	Recommended Value
defaultBigfileThreshold	Optional	Indicates the threshold for triggering multipart tasks, in bytes. If the size of a file to be uploaded, downloaded, or copied is greater than the threshold, the file is uploaded, downloaded, or copied in multiple parts. The default value is 50 MB. **NOTE** This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.	It is recommended that the value be greater than 5 MB.
defaultPartSize	Optional	Size of each part, in bytes. The default value is **auto**. **NOTE** ● For multipart upload and copy, the value ranges from 100 KB to 5 GB. ● For multipart download, the value is unrestricted. ● This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes. ● If this parameter is set to **auto**. In this case, obsutil automatically sets the part size for each multipart task based on the source object size.	[9MB, 100MB]
defaultParallels	Optional	Maximum number of concurrent tasks in the multipart mode. The default value is 5.	Set this parameter according to **8.4 Fine-Tuning obsutil Performance**.
defaultJobs	Optional	Maximum number of concurrent tasks in batches. The default value is 5. **NOTE** Batch tasks include uploading, downloading, and copying folders, as well as restoring and deleting objects in batches.	[1, 50]
defaultJobsCacheCount	Optional	Cache size of a batch task queue, indicating the maximum number of tasks that can be cached. The default value is 1000000. **NOTE** More cached tasks consume more memory resources. Therefore, you are advised to adjust the value of this parameter based on site requirements.	Default value

Parameter	Optional or Mandatory	Description	Recommended Value
rateLimitThreshold	Optional	Indicates the traffic control threshold of an upload or download request, in bytes. The default value is 0, indicating that traffic is not limited. The minimum value is 10 KB. **NOTE** This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.	It is recommended that the value be greater than 100 KB.
sdkLogBackups	Optional	Maximum number of SDK log files that can be retained. The default value is 10.	N/A
sdkLogLevel	Optional	SDK log level. Possible values are: ● DEBUG ● INFO ● WARN ● ERROR The default value is **WARN**.	N/A
sdkLogPath	Optional	Indicates the absolute path of SDK logs. The value must be a file path. The default value is the path of the **obssdk.log** file in the subfolder **obsutil_log** of the user's home directory (**HOME** in Linux or macOS and **C:\Users**<*Username*> in Windows). **NOTE** ● If this parameter is left blank, no SDK log is generated. ● The path must be a file path and cannot be a folder path. ● After the SDK log function is enabled, all logs of requests to OBS are saved in the SDK log file for problem analysis and location. ● Ensure that the user who runs the command has the read and write permissions on the path. **NOTICE** If multiple obsutil processes are running at the same time, log files may fail to be written concurrently or may be lost. In this case, add parameter **-config** when running commands to configure an independent configuration file for each process.	N/A

Parameter	Optional or Mandatory	Description	Recommended Value
sdkMaxLogSize	Optional	Size of an SDK log file, in bytes. The default value is 30 MB. **NOTE** This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.	The recommended value ranges from 10 MB to 100 MB.
utilLogBackups	Optional	Maximum number of obsutil log files that can be retained. The default value is 10.	N/A
utilLogLevel	Optional	obsutil log level. Possible values are: ● DEBUG ● INFO ● WARN ● ERROR The default value is **INFO**.	N/A
utilLogPath	Optional	Indicates the absolute path of obsutil logs. The value must be a file path. The default value is the path of the **obsutil.log** file in the subfolder **.obsutil_log** of the user's home directory (**HOME** in Linux or macOS and **C:\Users**\<*Username*> in Windows). **NOTE** ● If this parameter is left blank, no obsutil log is generated. ● The path must be a file path and cannot be a folder path. ● After the obsutil log function is enabled, all logs generated during commands executing are saved in the obsutil log file for problem analysis and location. ● Ensure that the user who runs the command has the read and write permissions on the path. **NOTICE** If multiple obsutil processes are running at the same time, log files may fail to be written concurrently. In this case, add parameter **-config** when running commands to configure an independent configuration file for each process.	N/A

Parameter	Optional or Mandatory	Description	Recommended Value
utilMaxLogSize	Optional	Size of an obsutil log file, in bytes. The default value is 30 MB. **NOTE** This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.	The recommended value ranges from 10 MB to 100 MB.
writeBufferIoSize	Optional	Size of the cache for downloading data, in bytes. The default value is 65536. **NOTE** - Set this parameter based on site requirements. If the size of the file to be downloaded is large, set this parameter to a large value. - This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.	N/A
readBufferIoSize	Optional	Size of the cache for uploading data, in bytes. The default value is 8192. **NOTE** - Set this parameter based on site requirements. If a large number of small files are uploaded, set this parameter to a small value. If large files are uploaded, set this parameter to a large value. - This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.	The recommended value ranges from 4096 to 65536.
recordMaxLogSize	Optional	Size of a result list containing success, failure, or warning lists in a batch task, in bytes. The default value is 30 MB. **NOTE** This value can contain a capacity unit. For example, **1 MB** indicates 1048576 bytes.	The recommended value ranges from 5 MB to 100 MB.
recordBackups	Optional	Maximum number of result lists of successful or failed batch tasks that can be retained. The default value is 1024.	N/A
humanReadableFormat	Optional	Indicates whether to convert the number of bytes in the object listing result and result list content to the human-readable format. The default value is **true**.	N/A

Parameter	Optional or Mandatory	Description	Recommended Value
showProgressBar	Optional	Indicates whether to display the progress bar on the console. The value **true** indicates that the progress bar is displayed. The default value is **true**.	N/A
showStartTime	Optional	Indicates whether to display the start time on the console. The value **true** indicates that start time is displayed. The default value is **true**.	N/A
colorfulProgress	Optional	Indicates whether to enable the progress bar with colors. The value **true** indicates that the bar is enabled.	N/A
helpLanguage	Optional	Language of the help documents. Options are as follows: ● Chinese ● English The default value is **English**.	N/A
defaultTempFileDir	Optional	Indicates the directory for storing temporary files during download. The default value is the **.obsutil_tempfile** subfolder in the user directory (**HOME** in Linux or macOS and **C:\Users** \<*Username*> in Windows). **NOTE** ● Temporary files generated during multipart download are stored in this directory. Therefore, ensure that the user who executes obsutil has the write permission on the path. ● The available space of the partition where the path is located must be greater than the size of the objects to be downloaded.	N/A
checkSourceChange	Optional	Indicates whether to check the change of source files or objects during upload/download/copy. The value **true** indicates that the function is enabled.	N/A

Parameter	Optional or Mandatory	Description	Recommended Value
skipCheckEmptyFolder	Optional	Indicates whether to skip checking empty folders on the OBS server during download. **true** indicates to skip the check. The default value is **false**. **NOTICE** If this parameter is set to **true**, the directory structure downloaded to your local PC may be different from that in OBS.	N/A
fsyncForDownload	Optional	Indicates whether to forcibly synchronize memory data to disks during download. The value **true** indicates to enable forcible synchronization. The default value is **false**. **NOTE** ● Set this parameter to **true** for scenarios that require high data reliability. ● If this parameter is set to **true**, the download performance will be deteriorated. Therefore, exercise caution when using this parameter.	N/A
memoryEconomicalScanForUpload	Optional	Indicates whether to use the scanning mode that occupies less memory space when uploading a folder. The value **true** indicates using this method and the default value **false** indicates not using this method.	N/A
forceOverwriteForDownload	Optional	Indicates to forcibly overwrite the local executable file (even if the local executable file is running) when downloading objects to the Linux OS or macOS. The value **true** means to overwrite, and the default value is **true**.	N/A
panicForSymbolicLinkCircle	Optional	Indicates the processing method after a symbolic link loop is detected during upload. The value **false** indicates that errors are only recorded. The value **true** indicates that panic is triggered. The default value is **false**.	N/A

Parameter	Optional or Mandatory	Description	Recommended Value
fastFailThreshold	Optional	Threshold for fast failure upon 4XX errors of batch tasks. When the number of 4XX errors exceeds the threshold, the fast failure process is triggered. All tasks that are not executed or being scanned are suspended. The default value is **5**. **NOTE** The fast failure mechanism is to avoid excessive traffic generated during batch task execution. To start a fast failure as soon as possible, set this parameter to **0** or **-1**, indicating that the fast failure process starts immediately whenever a 4XX error occurs.	N/A
abortHttpStatusForResumableTasks	Optional	HTTP status codes for fast interruption of multipart upload, download, and copy tasks. If a sub-task of a multipart task receives an HTTP code that falls into this range, the multipart task is immediately interrupted. The default values are 401, 403, 404, 405, and 409. **NOTE** - Multiple HTTP status codes can be carried and separated by commas (,). For example: 401,403,404; - The status code must be a 4XX HTTP status code. Other status codes are ignored.	Default value
showBytesForCopy	Optional	Indicates whether the progress bar displays the rate in bytes when objects are copied between buckets. The default value is **false**.	N/A
proxyUrl	Optional	HTTP proxy example: **http://username:password@your-proxy:8080** **NOTE** The user name and password cannot contain colons (:) and at signs (@), which will result in parsing errors.	N/A

NOTE

- Set parameters with **N/A** as the recommended value based on your needs.
- You are advised to specify **sdkLogPath** and **utilLogPath** to enable SDK logging and obsutil logging.
- The values of **defaultBigfileThreshold**, **defaultPartSize**, **rateLimitThreshold**, **sdkMaxLogSize**, **utilMaxLogSize**, **recordMaxLogSize**, **readBufferIoSize**, and **writeBufferIoSizecan** can contain a capacity unit. For example, 1 MB indicates 1048576 bytes.

B Change History

Release Date	What's New
2019-09-30	This issue is the first official release.