

对象存储服务

# 工具指南(obsutil) (安卡拉区域)

文档版本 01  
发布日期 2024-04-15



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

# 目录

<b>1 obsutil 简介</b>	<b>1</b>
<b>2 下载和安装 obsutil</b>	<b>4</b>
<b>3 快速入门</b>	<b>6</b>
3.1 创建访问密钥 (AK 和 SK)	6
3.2 初始化配置	6
3.3 快速使用	8
<b>4 桶相关命令</b>	<b>11</b>
4.1 创建桶	11
4.2 列举桶	12
4.3 查询桶属性	14
4.4 设置桶属性	15
4.5 删除桶	17
<b>5 对象相关命令</b>	<b>18</b>
5.1 创建文件夹	18
5.2 上传对象	19
5.3 查询对象属性	27
5.4 设置对象属性	29
5.5 列举对象	32
5.6 复制对象	36
5.7 移动对象	42
5.8 下载对象	48
5.9 生成对象的下载链接	54
5.10 删除对象	57
5.11 增量同步上传对象	59
5.12 增量同步复制对象	65
5.13 增量同步下载对象	70
5.14 恢复失败的上传任务	75
5.15 恢复失败的复制任务	79
5.16 恢复失败的下载任务	84
5.17 列举分段上传任务	89
5.18 删除分段上传任务	90

<b>6 辅助命令</b>	<b>93</b>
6.1 更新配置文件	93
6.2 删除断点记录文件	94
6.3 查看命令帮助	95
6.4 查看版本号	97
6.5 归档日志文件	98
6.6 列举失败结果清单文件	99
<b>7 常见示例</b>	<b>101</b>
7.1 上传示例	101
7.2 同步上传示例	103
7.3 下载示例	104
7.4 同步下载示例	105
7.5 复制示例	106
7.6 同步复制示例	107
7.7 列举示例	107
7.8 列举分段上传任务示例	108
7.9 删除桶内全部分段上传任务	108
<b>8 问题定位</b>	<b>110</b>
8.1 问题定位简介	110
8.2 日志文件	110
8.3 结果清单	111
8.4 返回码	112
<b>9 最佳实践</b>	<b>113</b>
9.1 利用 obsutil 帮助命令查找工具功能	113
9.2 利用 crontab 配置定时任务	114
9.3 将 obsutil 设为系统内置命令	115
9.4 obsutil 性能调优最佳实践	117
9.5 利用 obsutil 进行断点续传	118
9.6 利用 obsutil 上传软链接	119
9.7 为 obsutil 配置 HTTP 代理	119
9.8 使用 obsutil 实现客户端跨区域复制	119
9.9 指定 obsutil 上传下载时进行限速	120
<b>10 常见问题</b>	<b>121</b>
10.1 是否支持同步删除文件	121
10.2 列举接口是否支持保存结果到本地文件	121
10.3 列举桶内对象返回的大小与控制台桶内对象大小不一致	121
10.4 批量任务执行失败如何确认错误信息	122
10.5 i/o timeout、EOF 错误原因定位及解决	122
10.6 批量任务进度条中出现问号的原因	122
10.7 是否支持多个 config 文件在同一个目录下	123

---

<b>A 配置参数说明</b> .....	<b>124</b>
<b>B 修订记录</b> .....	<b>133</b>

# 1 obsutil 简介

obsutil是一款用于访问管理对象存储服务（Object Storage Service，OBS）的命令行工具，您可以使用该工具对OBS进行常用的配置管理操作，如创建桶、上传文件/文件夹、下载文件/文件夹、删除文件/文件夹等。对于熟悉命令程序的用户，obsutil能在执行批量处理、自动化任务场景能为您带来更优体验。

obsutil适用于Windows、Linux和macOS三大主流的操作系统，在各操作系统上推荐使用的版本如表1-1所示。您可以[单击这里](#)获取到obsutil在不同操作系统上的下载地址及下载方法。

表 1-1 obsutil 推荐使用的操作系统版本

操作系统	推荐使用版本
Windows	<ul style="list-style-type: none"><li>Windows 7</li><li>Windows 8</li><li>Windows 10</li><li>Windows 11</li><li>Windows Server 2016</li><li>Windows Server 2019</li></ul>
Linux	<ul style="list-style-type: none"><li>SUSE 11</li><li>Euler OS 2</li><li>CentOS 7</li></ul>
macOS	macOS 10.13.4

## 工具优势

obsutil具有四大优势：

1. 简单、易用；
2. 无需安装，轻便小巧，即下即用；
3. 同时支持Windows/Linux/macOS三大平台；

- 4. 配置多元化，性能优良。

## 使用场景

- 自动化备份归档等场景，比如定时上传本地数据至OBS。
- 支持通过OBS Browser+等其他工具不能实现的场景，比如同步上传/下载/复制对象等。

## 功能概述

obsutil提供的功能如表1-2所示。

表 1-2 obsutil 功能概述

功能	说明
<b>桶基本操作</b>	指定区域创建桶、删除桶以及获取桶列表、桶配置信息等。
<b>对象基本操作</b>	管理对象，包括上传、下载、删除和列举对象等。 <ul style="list-style-type: none"><li>• 支持上传单个或批量上传多个文件或文件夹。</li><li>• 支持分段上传大文件。</li><li>• 支持增量同步上传、下载和复制对象。</li><li>• 支持复制单个对象或按对象名前缀批量复制多个对象。</li><li>• 支持移动单个对象或按对象名前缀批量移动多个对象。</li><li>• 支持对失败的上传、下载、复制等任务进行恢复。</li></ul>
<b>日志记录</b>	支持在客户端配置日志记录，记录对桶和对象的操作日志，方便统计与分析。

obsutil目前暂不支持高级的桶管理和对象管理操作，相关功能会在后续版本中开发，如需配置，建议使用对象存储服务控制台。

## 命令行结构

遵循简单易用的原则，obsutil的命令行结构如下：

- Windows操作系统：  
**obsutil command [parameters...] [options...]**
- macOS/Linux操作系统：  
**./obsutil command [parameters...] [options...]**

## 📖 说明

- `command`为执行的命令，例如`ls`，`cp`等。
- `parameters`为该命令的基本参数（必选），例如创建桶时的桶名称。
- `options`为该命令的附加参数（通常为可选），且附加参数在运行命令时必须以“-”开头。
- 方括号[]不是命令的一部分，在输入命令时，参数不能使用方括号[]括起来。
- 如命令中含有特殊字符，如`&`、`<`、`>`以及空格等，则需要加引号转义（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）。
- 附加参数支持两种传入方式-`key=value`和-`key value`，例如-`acl=private`和-`acl private`。两种参数传入方式无区别，您可以根据使用习惯选择任意一种方式。
- 在Windows操作系统上，obsutil支持直接运行obsutil.exe进入交互命令模式。在该模式下输入命令时可以省略“obsutil”，直接输入**command [parameters...] [options...]**运行命令。示例如下：

```
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: 3

Input your command:
-->
```
- 以SSH远程登录macOS/Linux操作系统使用obsutil命令时，建议配置`TMOU=0`，防止SSH会话过期导致程序退出。



# 2 下载和安装 obsutil

## 下载地址

针对不同操作系统，obsutil的下载地址集成在OBS控制台上，登录OBS控制台，在控制台主界面单击obsutil的下载链接进行下载。

## 快速使用 obsutil

在不同操作系统，下载obsutil的方式也有所不同，下载后无需安装，即可开始使用。

### 说明

执行以下命令前要求云服务器已连接到互联网，否则安装obsutil客户端失败。

### Windows操作系统

- 步骤1** 通过控制台给出的下载链接将obsutil工具下载至本地。
- 步骤2** 下载完成后将其解压。
- 步骤3** 使用cmd进入该解压后的文件夹路径，执行obsutil命令。

----结束

### Linux操作系统

- 步骤1** 打开命令行终端，对控制台给出的下载链接执行wget命令将obsutil工具下载至本地  
Linux AMD 64位(Linux x86 64位)：

### 说明

- wget后的URL参数是安装包的下载网址，您无需适配，只需要复制并运行上方提供的命令，直接下载即可。
- 您也可以在一台Windows计算机上下载到软件包后，通过常见的跨平台传输工具（比如WinSCP）传输至您的Linux运行机。

- 步骤2** 在软件包所在目录，执行以下解压命令。

```
tar -xzf obsutil_linux_amd64.tar.gz
```

- 步骤3** 进入obsutil所在目录，执行以下命令，为obsutil增加可执行权限。

```
chmod 755 obsutil
```

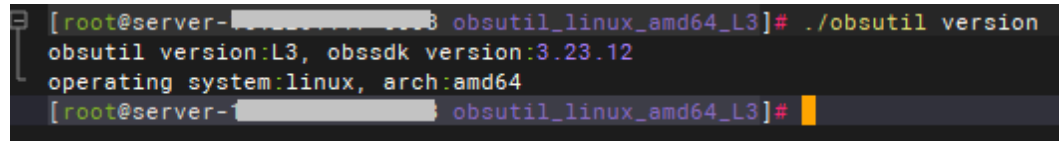
### 📖 说明

您必须为obsutil增加可执行权限，否则会在查询obsutil版本号时出现No such file or directory报错。

**步骤4** 继续在目录中执行以下命令，如果能顺利返回obsutil版本号，说明安装成功。

```
./obsutil version
```

图 2-1 安装成功



```
[root@server-3 obsutil_linux_amd64_L3]# ./obsutil version
obsutil version:L3, obssdk version:3.23.12
operating system:linux, arch:amd64
[root@server-3 obsutil_linux_amd64_L3]#
```

----结束

### macOS操作系统

**步骤1** 通过控制台给出的下载链接将obsutil工具下载至本地。

**步骤2** 下载完成后将其解压。

**步骤3** 打开命令行终端，进入obsutil所在目录，执行以下命令，为obsutil增加可执行权限。

```
chmod 755 obsutil
```

----结束

### 相关参考

- 如何快速使用obsutil请参见[快速使用](#)章节。
- 如何借助obsutil的命令行进一步熟悉工具使用方法，请参见[利用obsutil帮助命令查找工具功能](#)章节。

# 3 快速入门

## 3.1 创建访问密钥 (AK 和 SK)

在调用接口的时候，需要使用AK/SK进行签名验证。AK/SK获取步骤如下：

**步骤1** 登录控制台。

**步骤2** 单击界面右上角的登录用户名，在下拉列表中单击“我的凭证”。

**步骤3** 在左侧导航栏单击“访问密钥”。

**步骤4** 单击“新增访问密钥”，进入“新增访问密钥”页面。

**步骤5** 输入访问密钥描述信息（非必填），单击“确定”。

**步骤6** 通过手机短信、邮箱或者虚拟MFA进行验证，输入对应的验证码，单击“确定”。

### 📖 说明

如果您已开启操作保护，需要通过手机短信、邮箱或者虚拟MFA进行验证，输入对应的验证码。

**步骤7** 单击“立即下载”，浏览器下载访问密钥。

### 📖 说明

为防止访问密钥泄露，建议您将其保存到安全的位置。

----结束

## 3.2 初始化配置

使用obsutil之前，您需要配置obsutil与OBS的对接信息，包括OBS终端节点地址（Endpoint）和访问密钥（AK和SK）。获得OBS的认证后，才能使用obsutil执行OBS桶和对象的相关操作。

### 前提条件

- 已下载obsutil软件包，具体请参见[下载和安装obsutil](#)。

- 已获取OBS当前开通区域和终端节点地址，您可以从[这里](#)查看。
- 已获取访问密钥（AK和SK），具体操作请参见[创建访问密钥（AK和SK）](#)。

## 配置方法

您可以通过config命令对obsutil进行初始化配置（关于config的详细介绍，参见[更新配置文件](#)），示例如下：

### 📖 说明

- 以上命令运行后，会在运行obsutil命令的用户目录（macOS/Linux操作系统的“~”目录；Windows操作系统的“C:\Users\- .obsutilconfig的配置文件中的详细配置参数说明请参见[配置参数说明](#)。
- .obsutilconfig文件中保存有用户的AK和SK等信息，为避免密钥泄露，.obsutilconfig默认为隐藏文件，您可以在运行obsutil命令的用户目录下通过使用以下命令查看到该文件：
  - Windows操作系统  
dir
  - macOS/Linux操作系统  
ls -a或  
ls -al
- obsutil会对.obsutilconfig文件中的AK和SK进行加密以保护密钥安全。

- Windows操作系统

使用永久AK、SK进行初始化配置（token无需填写，回车跳过即可）：

```
obsutil config -interactive
```

```
Please input your ak:
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
Please input your sk:
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
Please input your endpoint:
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
Please input your token:
```

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

```
Update config file successfully!
```

使用临时AK、SK、SecurityToken进行初始化配置：

```
obsutil config -interactive
```

```
Please input your ak:
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
Please input your sk:
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
Please input your endpoint:
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
Please input your token:
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

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

```
Update config file successfully!
```

- macOS/Linux操作系统

使用永久AK、SK进行初始化配置（token无需填写，回车跳过即可）：

```
./obsutil config -interactive  
  
Please input your ak:  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Please input your sk:  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Please input your endpoint:  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Please input your token:  
  
Config file url:  
/root/.obsutilconfig  
  
Update config file successfully!
```

使用临时AK、SK、SecurityToken进行初始化配置：

```
./obsutil config -interactive  
  
Please input your ak:  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Please input your sk:  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Please input your endpoint:  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
Please input your token:  
xxxxxxxxxxxxxxxxxxxxxxxxxxxx  
  
Config file url:  
/root/.obsutilconfig  
  
Update config file successfully!
```

## 检查连通性

配置完成后，您可以通过如下方式检查连通性，确认配置是否无误。

- Windows操作系统  
obsutil ls -s
- macOS/Linux操作系统  
./obsutil ls -s

根据命令回显结果，检查配置结果：

- 如果返回结果中包含“Bucket number :”，表明配置正确。
- 如果返回结果中包含“Http status [403]”，表明访问密钥配置有误。
- 如果返回结果中包含“A connection attempt failed”，表明无法连接OBS服务，请检查网络环境是否正常。
- 如果返回结果中包含“Error: cloud\_url [url] is not in well format”，表明访问域名有误，请检查配置文件中域名是否正常。

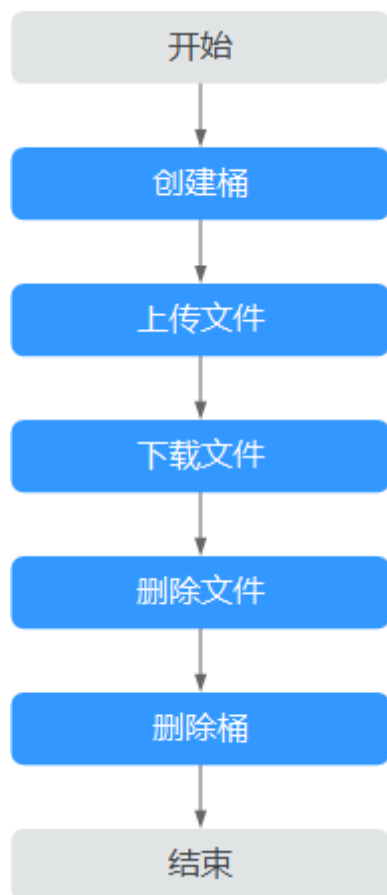
### 📖 说明

如果返回结果中包含“Http status [403]”，也可能是没有获取桶列表的权限，需要视具体场景进一步确认根因。

## 3.3 快速使用

本章节以Linux操作系统为例，介绍如何通过obsutil在OBS上完成基本的数据存取操作，具体操作流程如[图3-1](#)所示。

图 3-1 obsutil 快速使用流程



## 前提条件

- 已获取到obsutil工具，并完成了[初始化配置](#)。
- 已进入obsutil工具所在目录。

## 操作步骤

**步骤1** 运行`./obsutil mb obs://bucket-test`命令，创建一个名为bucket-test的新桶。

```
./obsutil mb obs://bucket-test  
Create bucket [bucket-test] successfully!
```

**步骤2** 运行`./obsutil cp /temp/test.txt obs://bucket-test/test.txt`命令，将本地test.txt文件上传至bucket-test桶中。

```
./obsutil cp /temp/test.txt obs://bucket-test/test.txt  
  
Parallel: 5          Jobs: 5  
Threshold: 52428800 PartSize: 5242880  
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
```

### 📖 说明

如果想要将本地test文件夹上传OBS桶中，对应命令是：

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

- cp上传命令详解请参考[上传对象](#)。
- 更多上传场景请参考[上传示例](#)。

**步骤3** 运行`./obsutil cp obs://bucket-test/test.txt /temp/test1.txt`命令，将bucket-test桶中的test.txt对象下载至本地。

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

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

### 📖 说明

如果想要将OBS桶内test目录下载至本地temp文件夹中，对应命令是：

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

- cp下载命令详解请参考[下载对象](#)。
- 更多下载场景请参考[下载示例](#)。

**步骤4** 运行`./obsutil rm obs://bucket-test/test.txt -f`命令，在bucket-test桶中删除test.txt对象。

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

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

**步骤5** 运行`./obsutil rm obs://bucket-test -f`命令，删除bucket-test桶。

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

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

----结束

# 4 桶相关命令

## 4.1 创建桶

### 功能说明

按照用户指定的桶名创建一个新桶。新创建桶的桶名在OBS中必须是唯一的。一个用户可以拥有的桶的数量不能超过100个。

#### 📖 说明

如果是同一个用户在同一区域的重复创建同名桶不会报错，返回HTTP状态码200，创建的桶属性以第一次请求为准。除此以外的其他场景重复创建同名桶返回HTTP状态码409，表明桶已存在。

### 命令结构

- Windows操作系统  
`obsutil mb obs://bucket [-fs] [-az=xxx] [-acl=xxx] [-location=xxx] [-config=xxx]`
- macOS/Linux操作系统  
`./obsutil mb obs://bucket [-fs] [-az=xxx] [-acl=xxx] [-location=xxx] [-config=xxx]`

### 运行示例

- 以Windows操作系统为例，运行**obsutil mb obs://bucket-test**，创建新桶成功。  

```
obsutil mb obs://bucket-test
Start at 2024-09-29 07:52:11.3769487 +0000 UTC

Create bucket [bucket-test] successfully, request id [000001923CC401864018BA75753D2D5F]
```
- 以Windows操作系统为例，运行**obsutil mb obs://bucket001**，创建其他账户下已存在的桶失败。  

```
obsutil mb obs://bucket001
Start at 2024-09-30 07:03:50.1378331 +0000 UTC

Create bucket [bucket001] failed, 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 and try again.], request id [0000019241BE18DB4019EDD66E135C56]
```



## 参数说明

参数	约束	描述
bucket	必选	桶名。 <b>说明</b> 桶命名规则如下： <ul style="list-style-type: none"><li>• 3~63个字符，数字或字母开头，支持小写字母、数字、“-”、“.”。</li><li>• 禁止使用类IP地址。</li><li>• 禁止以“-”或“.”开头及结尾。</li><li>• 禁止两个“.”相邻（如：“my..bucket”）。</li><li>• 禁止“.”和“-”相邻（如：“my-.bucket”和“my.-bucket”）。</li></ul>
fs	附加参数，可选	创建并行文件系统。
az	附加参数，可选	指定桶的数据冗余存储策略。支持的值： <ul style="list-style-type: none"><li>• multi-az</li></ul> <b>说明</b> multi-az代表创建数据冗余存储策略为多AZ存储的桶；不携带此参数代表创建数据冗余存储策略为单AZ存储的桶。
acl	附加参数，可选	创桶时可指定的预定义访问策略。支持的值： <ul style="list-style-type: none"><li>• private</li><li>• public-read</li><li>• public-read-write</li></ul> <b>说明</b> 以上三个值分别对应：私有读写、公共读、公共读写。
location	附加参数，如果请求的OBS服务地址所在区域为默认区域，则可为空，否则为必选	桶所在的区域。
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考。

## 4.2 列举桶

### 功能说明

获取桶列表，查询到的桶列表将以桶名字典序排列。

### 命令结构

- Windows操作系统  
obsutil ls [-s] [-limit=1] [-format=default] [-config=xxx]
- macOS/Linux操作系统

```
./obsutil ls [-s] [-du] [-fs] [-j=1] [-limit=1] [-format=default] [-config=xxx]
```

## 运行示例

- 以Windows操作系统为例，运行**obsutil ls -limit=5**，查询桶列表。

```
obsutil ls -limit=5
Start at 2024-09-29 07:58:46.0506904 +0000 UTC

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 : 5
```

## 参数说明

参数	约束	描述
s	附加参数，可选	以精简格式显示查询结果。 <b>说明</b> 在精简格式下，返回结果只包含桶名。
du	附加参数，可选	快速返回所列举对象的总大小，不显示具体的对象信息，可与其他参数配合使用。 <b>说明</b> 此参数在列举桶时无效，仅在列举对象时生效。
limit	附加参数，可选	查询桶的最大个数，小于0表示列举所有结果，不设置时的默认值为1000。
format	附加参数，可选	指定以自定义格式打印列举结果。当前仅支持值[default]，指定列举结果在一行显示。
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考。 。

### 说明

列举桶的结果中，BucketType字段表示桶的类型，OBJECT代表对象桶；POSIX代表并行文件系统。

## 4.3 查询桶属性

### 功能说明

查询单个桶的基本属性，包括桶的区域、桶的版本号、桶是否支持文件接口、桶的可用区、桶中对象数量、桶的存储用量以及桶的配额。

### 命令结构

- Windows操作系统  
`obsutil stat obs://bucket [-acl] [-bf=xxx] [-config=xxx]`
- macOS/Linux操作系统  
`./obsutil stat obs://bucket [-acl] [-bf=xxx] [-config=xxx]`

### 运行示例

- 以Windows操作系统为例，运行**`obsutil stat obs://bucket-test`**，查询bucket-test桶的基本属性。

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

Start at 2024-09-29 07:58:46.0506904 +0000 UTC

Bucket:
  obs://bucket-test
Location:
  your-location
ObsVersion:
  3.0
AvailableZone:
  multi-az
BucketType:
  OBJECTObjectNumber:
  8005
Size:
  320076506
Quota:
  0
```

### 参数说明

参数	约束	描述
bucket	必选	桶名。
acl	可选	同时查询桶的访问策略。
bf	附加参数，可选	桶的存储用量字节数的显示格式。支持的值： <ul style="list-style-type: none"><li>human-readable</li><li>raw</li></ul> <b>说明</b> 如果未设置该参数，则桶的存储用量字节数的显示格式由配置文件中的humanReadableFormat参数决定。
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考。

## 响应结果

字段名	描述
Bucket	桶名。
Location	桶的区域位置。
ObsVersion	桶的版本。
AvailableZone	桶的可用区。
BucketType	桶的类型，OBJECT代表对象桶；POSIX代表并行文件系统。
ObjectNumber	桶内对象个数。
Size	桶的存储用量，单位：字节。
Quota	桶的配额值，0代表配额没有上限。
Acl	桶的访问策略。

## 4.4 设置桶属性

### 功能说明

设置桶的访问策略等属性。

### 命令结构

- Windows操作系统  
`obsutil chatter obs://bucket [-acl=xxx] [-aclXml=xxx] [-config=xxx]`
- macOS/Linux操作系统  
`./obsutil chatter obs://bucket [-acl=xxx] [-aclXml=xxx] [-config=xxx]`

### 运行示例

- 以Windows操作系统为例，运行**obsutil chatter obs://bucket-test -acl=private**，修改桶的访问策略为私有读写。  
`obsutil chatter obs://bucket-test -acl=private`  
Start at 2024-09-29 07:58:46.0506904 +0000 UTC  
Set the acl of bucket [bucket-test] to [private] successfully, request id [04050000016836C5DA6FB21F14A2A0C0]

### 参数说明

参数	约束	描述
bucket	必选	桶名。

参数	约束	描述
acl	附加参数, 可选	桶的预定义访问策略。支持的值: <ul style="list-style-type: none"> <li>• private</li> <li>• public-read</li> <li>• public-read-write</li> </ul> <b>说明</b> 以上三个值分别对应: 私有读写、公共读、公共读写, 三种预定义访问策略。
aclX ml	附加参数, 可选	桶的访问策略 (xml格式表示)。 <pre>&lt;AccessControlPolicy&gt;   &lt;Owner&gt;     &lt;ID&gt;ownerid&lt;/ID&gt;   &lt;/Owner&gt;   &lt;AccessControlList&gt;     &lt;Grant&gt;       &lt;Grantee&gt;         &lt;ID&gt;userid&lt;/ID&gt;       &lt;/Grantee&gt;       &lt;Permission&gt;[WRITE WRITE_ACP READ READ_ACP  FULL_CONTROL]&lt;/Permission&gt;     &lt;/Grant&gt;     &lt;Grant&gt;       &lt;Grantee&gt;         &lt;Canned&gt;Everyone&lt;/Canned&gt;       &lt;/Grantee&gt;       &lt;Permission&gt;[WRITE WRITE_ACP READ READ_ACP  FULL_CONTROL]&lt;/Permission&gt;     &lt;/Grant&gt;   &lt;/AccessControlList&gt; &lt;/AccessControlPolicy&gt;</pre> <b>说明</b> <ul style="list-style-type: none"> <li>• Owner节点为可选, 指定桶拥有者ID。</li> <li>• AccessControlList节点下的Grant子节点包含被授权的用户列表, Grantee子节点指定被授权用户的ID, Canned子节点指定被授权的用户组 (当前仅支持Everyone)。</li> <li>• 可支持被授予的权限有: WRITE (写权限)、WRITE_ACP (写ACL权限)、READ (读权限)、READ_ACP (读ACL权限) 和FULL_CONTROL (完全控制权限)。</li> </ul> <b>须知</b> 由于该参数值必定含有<和>, 作为命令运行时, 必须使用引号转义 (macOS/Linux操作系统使用单引号, Windows操作系统使用双引号)。
confi g	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

### 📖 说明

每次命令只能选择设置acl和aclXml中的一个。

## 4.5 删除桶

### 功能说明

删除桶，待删除的桶必须为空（不包含对象、历史版本对象或分段上传碎片）。

#### 说明

如果要删除非空桶，请先使用[删除分段上传任务](#)命令和[删除对象](#)命令，将桶清空后再执行删除桶命令。

### 命令结构

- Windows操作系统  
`obsutil rm obs://bucket [-f] [-config=xxx]`
- macOS/Linux操作系统  
`./obsutil rm obs://bucket [-f] [-config=xxx]`

### 运行示例

- 以Windows操作系统为例，运行**`obsutil rm obs://bucket-test`**，删除bucket-test桶。  

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

Start at 2024-09-30 07:58:33.736622 +0000 UTC

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

### 参数说明

参数	约束	描述
bucket	必选	桶名。
f	附加参数，可选	强制操作，不进行询问提示。
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考。

# 5 对象相关命令

## 5.1 创建文件夹

### 功能说明

在指定桶内或本地文件系统中创建文件夹。

#### 须知

创建重名文件夹，不会报错，且文件夹内已存在的内容不发生变化。

### 命令结构

- Windows操作系统
  - 在指定桶内创建文件夹  
`obsutil mkdir obs://bucket/folder[/subfolder1/subfolder2] [-config=xxx]`
  - 在本地文件系统路径中创建文件夹  
`obsutil mkdir folder_url [-config=xxx]`
- macOS/Linux操作系统
  - 在指定桶内创建文件夹  
`./obsutil mkdir obs://bucket/folder[/subfolder1/subfolder2] [-config=xxx]`
  - 在本地文件系统路径中创建文件夹  
`./obsutil mkdir folder_url [-config=xxx]`

### 运行示例

- 以Windows操作系统为例，运行**`obsutil mkdir obs://bucket-test/folder1/folder2`**，在桶内创建文件夹。

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

The bucket [bucket-test] does not support POSIX, create folder(s) step by step
Create folder [obs://bucket-test/folder1/] successfully, request id
[0000016979E1D23C860BB3D8E4577C5E]
Create folder [obs://bucket-test/folder1/folder2] successfully, request id
[0000016979E1D2B2860BB5181229C72C]
```

## 参数说明

参数	约束	描述
bucket	在指定的桶内创建文件夹时必选	桶名。
folder	在指定的桶内创建文件夹时必选	桶内的文件夹路径，支持多级文件夹，以“/”分隔。
folder_url	在本地文件系统路径中创建文件夹时必选	本地文件系统中的文件夹路径，支持绝对路径和相对路径。
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考。

## 5.2 上传对象

### 功能说明

上传单个或多个本地文件或文件夹至OBS指定路径。待上传的文件可以是任何类型：文本文件、图片、视频等等。

#### 须知

上传本地文件或文件夹时，请确保本地文件或文件夹在上传过程不发生变化，否则可能产生上传失败或者数据不一致的问题。

### 约束与限制

obsutil对上传的文件或文件夹有大小限制，最小可以上传**0Byte**的空文件或文件夹，最大可以上传**5GB**（未采用分段上传）或**48.8TB**（采用分段上传）的单个文件。

### 命令结构

- Windows操作系统

- 上传文件

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

- 上传文件夹

```
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] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
```

- 多文件/文件夹上传

```
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] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-  
timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
```

### 说明

命令中的“/prefix”即为上传文件夹时的对象名前缀。运行示例请参见。

- Linux/Mac操作系统

- 上传文件

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

- 上传文件夹

```
./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] [-meta=aaa:bbb#ccc:ddd] [-  
ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-  
cpd=xxx] [-config=xxx]
```

- 多文件/文件夹上传

```
./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] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx][-  
timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
```

### 说明

命令中的“/prefix”即为上传文件夹时的对象名前缀。运行示例请参见。

## 运行示例

- 以Windows操作系统为例，运行**obsutil cp d:\temp\test.txt obs://bucket-test/key**，将D盘temp路径下名为test.txt的文件上传至bucket-test桶中，并且重命名为key。

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

```
Start at 2024-09-30 08:11:41.6724827 +0000 UTC
```

```
Parallel: 5          Jobs: 5  
Threshold: 50.00MB  PartSize: auto  
VerifyLength: false  VerifyMd5: false  
CheckpointDir: C:\Users\Administrator\obsutil_checkpoint
```

```
[=====] 100.00% 1.68 MB/s 8.46MB/  
8.46MB 5s
```

```
Upload successfully, 8.46MB, n/a, d:\temp\test.txt --> obs://bucket-test/key, cost [5], status [200],  
request id [0000016979E1D2B2860BB5181229C72C]
```

- 以Windows操作系统为例，运行**obsutil cp d:\temp obs://bucket-test -f -r**，递归上传D盘temp文件夹中的所有文件和子文件夹至bucket-test桶的temp文件夹中。

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

```
Start at 2024-09-30 08:14:12.1406275 +0000 UTC
```

```
Parallel: 5          Jobs: 5  
Threshold: 50.00MB  PartSize: auto  
VerifyLength: false  VerifyMd5: false  
CheckpointDir: C:\Users\Administrator\obsutil_checkpoint
```

```
Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed  
OutputDir: C:\Users\Administrator\obsutil_output
```

```
[=====] 100.00% tps:35.71 2.02 KB/s  
7.20MB/7.20MB 0s
```

```
Succeed count: 5    Failed count: 0
```

```
Succeed bytes: xxx
```

```
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71, transferred size:  
7.20MB]
```

Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed

- 更多运行示例，请参见。

## 参数说明

参数	约束	描述
file_url	多文件/文件夹上传时可选 上传文件时必选	本地文件路径。 <b>说明</b> <ul style="list-style-type: none"><li>• 多文件/文件夹上传时，禁止路径嵌套，例如：/a/b/c和/a/b/。</li><li>• 多文件/文件夹上传时，必须配置msm=1，此时可以用英文逗号分隔多个文件，例如：file_url1,file_url2。</li><li>• 多文件/文件夹上传时，文件和文件夹可以同时存在，例如：file_url1,folder_url1,file_url2,folder_url2。</li></ul>
folder_url	多文件/文件夹上传时可选 上传文件夹时必选	本地文件夹路径。 <b>说明</b> <ul style="list-style-type: none"><li>• 上传文件夹时，如果未设置flat选项，则直接上传整个文件夹；如果设置了flat选项，则上传文件夹下的所有内容。</li><li>• 多文件/文件夹上传时，禁止路径嵌套，例如：/a/b/c和/a/b/。</li><li>• 多文件/文件夹上传时，必须配置msm=1，此时可以用逗号分隔多个文件夹，例如：folder_url1,folder_url2。</li><li>• 多文件/文件夹上传时，文件和文件夹可以同时存在，例如：file_url1,folder_url1,file_url2,folder_url2。</li></ul>
filelist_url	多文件/文件夹上传时可选	包含待上传文件/文件夹列表的文件，必须配置msm=2。 <b>说明</b> <ul style="list-style-type: none"><li>• 该文件为普通文本文件（如.txt文件、.csv文件），文件中的每行代表一个待上传的文件/文件夹，例如： file_url1 file_url2 folder_url1 folder_url2</li><li>• 文件中待上传的文件/文件夹禁止路径嵌套，例如：/a/b/c和/a/b/。</li></ul>
bucket	必选	桶名。

参数	约束	描述
key	可选	<p>上传文件时的对象名/对象名前缀，或上传文件夹时的对象名前缀。</p> <p>规则如下：</p> <ul style="list-style-type: none"> <li>上传文件时，如果该值为空，则将对象上传到桶的根目录，对象名为文件名。如果该值不为空，如果该值以“/”结尾，则以该值作为对象名前缀上传对象，对象名为该值+文件名；否则，将该值作为对象名进行上传；</li> <li>上传文件夹时，如果该值为空，则将文件夹上传到桶的根目录；如果该值不为空，如果该值以“/”结尾，则将该值作为对象名前缀上传文件夹；否则，将文件夹上传到以该值加上“/”结尾的对象名前缀下。</li> </ul> <p><b>说明</b> 请参考理解该参数的使用方式。</p>
fr	附加参数，上传文件时可选	上传文件时生成结果清单文件。
flat	附加参数，多文件/文件夹上传或上传文件夹时可选	上传文件夹时，只上传该文件夹下的所有内容。
arcDir	附加参数，可选	上传文件成功后的归档路径，上传成功后的文件会移动到该路径下。
dryRun	附加参数，可选	测试模式运行，不执行实际的上传操作。
link	附加参数，可选	<p>上传软链接文件/文件夹指向的真实路径。</p> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>如果未指定该参数，而待上传的文件是一个软链接且该软链接指向的目标文件不存在，则Windows操作系统会产生“The system cannot find the file specified”的上传失败异常，而macOS/Linux操作系统会产生“No such file or directory”的上传失败异常。</li> <li>文件夹软链接不能形成环，否则上传会以panic的形式退出；如果不希望产生panic，请在配置文件中配置panicForSymbolicLinkCircle为false。</li> </ul>
u	附加参数，可选	增量上传操作，设置该参数后，上传每个文件时会对比桶中对应路径的对象，仅在对象不存在，或者对象大小与文件大小不一致，或者对象的最后修改时间早于文件的最后修改时间时进行上传。
vlength	附加参数，可选	上传完成后，验证桶中对象的大小是否与本地文件大小一致。

参数	约束	描述
vmd5	附加参数, 可选	上传完成后, 验证桶中对象的MD5值是否与本地文件的MD5值一致。 <b>说明</b> <ul style="list-style-type: none"><li>如果待上传的本地文件较大, 使用该参数将会因为计算MD5而导致整体性能下降。</li><li>MD5值校验通过后, 会将该值设置为对象元数据x-obs-meta-md5chksum, 用于下载或复制时校验MD5。</li></ul> <b>注意</b> <p>如果您的业务涉及加密场景, 请勿使用该参数。</p>
p	附加参数, 可选	每个分段上传任务的最大并发数, 默认为配置文件中的defaultParallels。
threshold	附加参数, 可选	开启分段上传任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"><li>如果待上传的文件小于该阈值, 则使用直接上传模式; 否则使用分段上传模式。</li><li>直接上传模式不会产生断点记录文件, 不支持断点续传。</li><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
acl	附加参数, 可选	上传文件时可指定的预定义访问策略。支持的值: <ul style="list-style-type: none"><li>private</li><li>public-read</li><li>public-read-write</li><li>bucket-owner-full-control</li></ul> <b>说明</b> <p>以上四个值分别对应: 私有读写、公共读、公共读写、桶拥有者完全控制, 四种预定义访问策略。</p>
meta	附加参数, 可选	上传文件时可指定的标准和自定义元数据。格式为: key1:value1#key2:value2#key3:value3。 <b>说明</b> <ol style="list-style-type: none"><li>以上值表示复制对象完成后, 桶内目标对象包含三组自定义元数据, 分别为: key1:value1、key2:value2、key3:value3。</li><li>标准元数据头包括: Content-Type、Content-Encoding、Cache-Control、Content-Disposition、Content-Language、Expires。</li></ol>

参数	约束	描述
ps	附加参数, 可选	每个分段上传任务的段大小, 单位: 字节, 取值范围是100KB~5GB, 默认为配置文件中的defaultPartSize。 <b>说明</b> <ul style="list-style-type: none"><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>支持配置为auto, 此时obsutil会根据源文件大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。 <b>说明</b> <p>每个分段上传任务会产生唯一对应的断点记录文件并保存至该文件夹的upload子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
r	附加参数, 上传文件夹时必选 多文件上传时可选	递归上传文件夹中的所有文件和子文件夹。
f	附加参数, 多文件/文件夹上传或上传文件夹时可选	强制操作, 不进行询问提示。
j	附加参数, 多文件/文件夹上传或上传文件夹时可选	上传文件夹时批量任务的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> <p>工具会保证该值至少为1。</p>
msm	附加参数, 多文件/文件夹上传时必选	开启多文件/文件夹上传模式, 支持的值: [1 2]。 <b>说明</b> <ul style="list-style-type: none"><li>如果msm=1则代表上传的URL是一组文件/文件夹列表(以英文逗号分隔)。</li><li>如果该值为msm=2则代表上传的URL是一个包含文件/文件夹列表的文件。</li><li>如果文件/文件夹名本身包含英文逗号, 请不要使用msm=1的模式。</li><li>如果没有设置r参数, 则列表中的文件夹不会被上传。</li></ul>

参数	约束	描述
exclude	附加参数，多文件/文件夹上传或上传文件夹时可选	<p>不包含文件的匹配模式，如：*.txt。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持“*”匹配多个任意字符和“?”匹配单个任意字符，例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li> <li>您可以使用“\*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。</li> <li>如果待上传的文件名匹配该参数，则跳过该文件的上传。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式（Linux/Mac操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。</li> <li>该匹配模式作用于文件全路径（含文件名和文件目录）。</li> <li>该匹配模式仅对文件夹中的文件生效。</li> <li>支持指定多个exclude参数，如 -exclude=*.xxx -exclude=*.xxx</li> </ul>
include	附加参数，多文件/文件夹上传或上传文件夹时可选	<p>包含文件的匹配模式，如：*.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持“*”匹配多个任意字符和“?”匹配单个任意字符。</li> <li>您可以使用“\*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。</li> <li>首先执行exclude的匹配规则，如果待上传的文件名不匹配exclude，则判断待上传的文件名是否匹配该参数，如果匹配则上传该文件，否则跳过该文件的上传。</li> </ul> <p><b>说明</b> 如何使用include实现前缀匹配上传： ./obsutil cp /localpath/ obs://test/ -include=/localpath/2022-12-09/* -f -r</p> <p>以上命令的含义是，上传本地路径localpath下符合匹配规则的文件到test桶下，匹配规则是以/localpath/2022-12-09/开头的。</p> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式（Linux/Mac操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。</li> <li>该匹配模式作用于文件全路径（含文件名和文件目录）。</li> <li>该匹配模式仅对文件夹中的文件生效。</li> <li>支持指定多个include参数，如 -include=*.xxx -include=*.xxx</li> </ul>
at	附加参数，多文件/文件夹上传或上传文件夹时可选	<p>指定按照文件最后访问时间作为timeRange匹配模式，仅上传最后访问时间满足timeRange时间段的文件。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>需配合timeRange使用</li> </ul>

参数	约束	描述
disableDirObject	附加参数, 多文件夹上传时可选	上传文件夹时候, 文件夹本身不会作为一个对象上传, 在有有很多空文件夹场景时候可以避免无用文件夹上传到对象存储桶中; 文件夹下有文件时候, 文件夹下文件依然会上传并保持原有路径格式, 不受影响。
timeRange	附加参数, 多文件/文件夹上传或上传文件夹时可选	<p>上传文件时的时间段匹配模式, 仅上传最后修改时间在该时间段内的文件。</p> <p>该匹配模式优先级低于文件名匹配模式: exclude和include, 优先执行文件名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>使用“time1-time2”, 代表匹配的时间段, 其中time1必须小于等于time2, 格式为yyyyMMddHHmmss。</li> <li>时间格式支持自动补全, 例如yyyyMMdd等价于yyyyMMdd000000, yyyyMM等价于yyyyMM01000000。</li> <li>使用“*-time2”, 代表匹配最后修改时间在time2之前的所有文件, 使用“time1-*”, 代表匹配最后修改时间time1之后的所有文件。</li> </ul> <p><b>须知</b> 该匹配模式表示的时间是UTC时间。</p>
mf	附加参数, 可选	设置名称匹配模式(include、exclude)和时间匹配模式(timeRange)对文件夹也生效。
o	附加参数, 可选	<p>生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件(可能包含成功结果、失败结果、警告结果三个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>结果清单文件命名规则: cp_{succeed   failed   warning}_report_时间_TaskId.txt。</li> <li>单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li> <li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件"cp_failed_report_时间_TaskId.txt"及日志路径下的。</li> </ul>
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

字段名	描述
Parallel	请求中的-p参数。
Jobs	请求中的-j参数。

字段名	描述
Threshold	请求中的-threshold参数。
PartSize	请求中的-ps参数。
Exclude	请求中的-exclude参数。
Include	请求中的-include参数。
TimeRange	请求中的-timeRange参数。
VerifyLength	请求中的-vlength参数。
VerifyMd5	请求中的-vmd5参数。
CheckpointDir	请求中的-cpd参数。
OutputDir	请求中的-o参数。
ArcDir	请求中的-arcDir参数。
Succeed count	成功的任务数。
Failed count	失败的任务数。
Skip count	增量上传/下载/复制、同步上传/下载/复制跳过的任务数。 <b>说明</b> 跳过的任务数也会记录到成功的任务数中。
Warning count	执行完成但包含警告的任务数。 <b>说明</b> <ul style="list-style-type: none"><li>产生警告的任务可能是失败的也可能是成功的，需要根据成功或失败的结果清单进一步判断。</li><li>产生警告的任务数与成功/失败任务数是相互独立的，总任务数仍是成功的任务数+失败的任务数。</li></ul>
Succeed bytes	上传/下载成功的字节数。
max cost	所有任务中的最长耗时，单位：毫秒。
min cost	所有任务中的最短耗时，单位：毫秒。
average cost	平均耗时，单位：毫秒。
average tps	平均每秒的任务完成个数。
Task id	每次运行的唯一标识号，用于查找批量任务生成的结果清单文件。

## 5.3 查询对象属性

### 功能说明

查询对象的基本属性。



## 命令结构

- Windows操作系统  
`obsutil stat obs://bucket/key [-acl][-bf=xxx] [-config=xxx]`
- macOS/Linux操作系统  
`./obsutil stat obs://bucket/key [-acl][-bf=xxx] [-config=xxx]`

## 运行示例

- 以Windows操作系统为例，运行**`obsutil stat obs://bucket-test/key`**，查询对象基本属性。

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

Start at 2024-09-25 04:48:10.1147483 +0000 UTC

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

## 参数说明

参数	约束	描述
bucket	必选	桶名。
key	必选	对象名。
acl	可选	同时查询对象的访问策略。
bf	附加参数，可选	对象大小字节数的显示格式。支持的值： <ul style="list-style-type: none"><li>human-readable</li><li>raw</li></ul> <b>说明</b> 如果未设置该参数，则对象大小字节数的显示格式由配置文件中的humanReadableFormat参数决定。
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考。

## 响应结果

字段名	描述
Key	对象名。
LastModified	对象的最后修改时间。

字段名	描述
Size	对象的大小，单位：字节。
MD5	对象真实的MD5值。 <b>说明</b> 仅当使用cp命令并且配置了-vmd5选项上传对象后，才能查询到该值。
ETag	服务端计算的对象ETag值。
ContentType	对象的Content-Type。
Type	对象的类型。
Metadata	对象的自定义元数据。 <b>说明</b> 对象有自定义元数据后，才能查询到该值。

## 5.4 设置对象属性

### 功能说明

设置对象的属性或按对象名前缀批量设置对象的属性。

### 命令结构

- Windows操作系统
  - 设置单个对象属性  
`obsutil chatter obs://bucket/key [-meta=aaa:bbb#ccc:ddd] [-acl=xxx] [-aclXml=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]`
  - 批量设置对象属性  
`obsutil chatter obs://bucket[/key] -r [-f] [-v] [-meta=aaa:bbb#ccc:ddd] [-acl=xxx] [-aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx]`
- macOS/Linux操作系统
  - 设置单个对象属性  
`./obsutil chatter obs://bucket/key [-meta=aaa:bbb#ccc:ddd] [-acl=xxx] [-aclXml=xxx] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]`
  - 批量设置对象属性  
`./obsutil chatter obs://bucket[/key] -r [-f] [-v] [-meta=aaa:bbb#ccc:ddd] [-acl=xxx] [-aclXml=xxx] [-o=xxx] [-j=1] [-config=xxx]`

### 运行示例

- 以Windows操作系统为例，运行**obsutil chatter obs://bucket-test/key -acl=public-read**，设置对象的访问策略为公共读。  
`obsutil chatter obs://bucket-test/key -acl=public-read`  
Start at 2024-09-30 08:17:15.5572914 +0000 UTC  
  
Set the acl of object [key] in the bucket [bucket-test] to [public-read] successfully, request id [04050000016836DDFA73B2B5320E2651]
- 以Windows操作系统为例，运行**obsutil chatter obs://bucket-test -r -f -acl=public-read**，批量设置桶内所有对象的访问策略为公共读。  
`obsutil chatter obs://bucket-test -r -f -acl=public-read`  
Start at 2024-09-30 08:18:03.105373 +0000 UTC

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

## 参数说明

参数	约束	描述
bucket	必选	桶名。
key	设置单个对象属性时必选 批量设置对象属性时可选	待设置属性的对象名，或批量设置对象属性时的对象名前缀。 <b>说明</b> 批量设置对象属性时，如果该值为空，则表示设置桶内所有对象的属性。
meta	附加参数，可选	复制对象时可指定的目标对象的标准和自定义元数据。 格式为：key1:value1#key2:value2#key3:value3。 <b>说明</b> 1. 该参数需要与direct搭配使用才会生效。 2. 以上值表示复制对象完成后，桶内目标对象包含三组自定义元数据，分别为：key1:value1、key2:value2、key3:value3。 3. 标准元数据头包括：Content-Type、Content-Encoding、Cache-Control、Content-Disposition、Content-Language、Expires。
direct	附加参数，可选	元数据操作指示符。 取值为REPLACE_NEW或REPLACE。 REPLACE_NEW表示：对于已经存在值的元数据进行替换，不存在值的元数据进行赋值，未指定的元数据保持不变。 REPLACE表示：使用当前请求中携带的头域完整替换，未指定的元数据会被删除。 <b>说明</b> 该参数需要与meta搭配使用才会生效。
acl	附加参数，可选	对象的预定义访问策略。支持的值： <ul style="list-style-type: none"> <li>• private</li> <li>• public-read</li> <li>• public-read-write</li> <li>• bucket-owner-full-control</li> </ul> <b>说明</b> 以上四个值分别对应：私有读写、公共读、公共读写、桶拥有者完全控制，四种预定义访问策略。

参数	约束	描述
aclxml	附加参数, 可选	<p>桶的访问策略 (xml格式表示)。</p> <pre>&lt;AccessControlPolicy&gt;   &lt;Owner&gt;     &lt;ID&gt;ownerid&lt;/ID&gt;   &lt;/Owner&gt;   &lt;AccessControlList&gt;     &lt;Grant&gt;       &lt;Grantee&gt;         &lt;ID&gt;userid&lt;/ID&gt;       &lt;/Grantee&gt;       &lt;Permission&gt;[WRITE WRITE_ACP READ READ_ACP FULL_CONTROL]&lt;/Permission&gt;     &lt;/Grant&gt;     &lt;Grant&gt;       &lt;Grantee&gt;         &lt;Canned&gt;Everyone&lt;/Canned&gt;       &lt;/Grantee&gt;       &lt;Permission&gt;[WRITE WRITE_ACP READ READ_ACP FULL_CONTROL]&lt;/Permission&gt;     &lt;/Grant&gt;   &lt;/AccessControlList&gt; &lt;/AccessControlPolicy&gt;</pre> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>Owner节点为可选, 指定对象拥有者ID。</li> <li>AccessControlList节点下的Grant子节点包含被授权的用户列表, Grantee子节点指定被授权用户的ID, Canned子节点指定被授权的用户组 (当前仅支持Everyone)。</li> <li>可支持被授予的权限有: WRITE (写权限)、WRITE_ACP (写ACL权限)、READ (读权限)、READ_ACP (读ACL权限) 和FULL_CONTROL (完全控制权限)。</li> </ul> <p><b>须知</b></p> <p>由于该参数值必定含有&lt;和&gt;, 作为命令运行时, 必须使用引号转义 (macOS/Linux操作系统使用单引号, Windows操作系统使用双引号)。</p>
versionid	附加参数, 设置单个对象属性时可选	待设置属性对象的版本号。
fr	附加参数, 设置单个对象属性时可选	设置单个对象属性时生成结果清单文件。
f	附加参数, 批量设置对象属性时可选	强制操作, 不进行询问提示。
r	附加参数, 批量设置对象属性时必选	按指定的对象名前缀批量设置对象属性。
v	附加参数, 批量设置对象属性时可选	按指定的对象名前缀批量设置多版本对象属性。

参数	约束	描述
o	附加参数, 可选	生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件 (包含成功结果和失败结果两个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"><li>结果清单文件命名规则: chattri_{succeed   failed}_report_时间_TaskId.txt。 单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件 "chattri_failed_report_时间_TaskId.txt"及日志路径下的。</li></ul>
j	附加参数, 批量设置对象属性时可选	批量设置对象属性的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> 工具会保证该值至少为1。
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

#### 📖 说明

每次命令只能选择设置acl和aclXml中的一个。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.5 列举对象

### 功能说明

查询桶内对象或多版本对象, 返回对象列表将按照对象名和版本号以字典序排列。

### 命令结构

- Windows操作系统  
obsutil ls obs://bucket[/prefix] [-s] [-d][-fs] [-v] [-du] [-marker=xxx] [-versionIdMarker=xxx] [-bf=xxx] [-limit=1] [-format=default] [-config=xxx]
- macOS/Linux操作系统  
./obsutil ls obs://bucket[/prefix] [-s] [-fs] [-d] [-v] [-du] [-marker=xxx] [-versionIdMarker=xxx] [-bf=xxx] [-limit=1] [-format=default] [-config=xxx]

### 运行示例

- 示例1: 以Windows操作系统为例, 运行**obsutil ls obs://bucket-test -limit=10**, 查询桶内对象。

```
obsutil ls obs://bucket-test -limit=10
Start at 2024-09-30 08:18:50.0592788 +0000 UTC

Listing objects .

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

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

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

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

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

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

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

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

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

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

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

- 示例2：以Windows操作系统为例，运行**obsutil ls obs://bucket-test2**，查询桶内对象。

```
obsutil ls obs://bucket-test2
Start at 2024-09-30 08:21:06.6300221 +0000 UTC

Listing objects .

Object list:
key                               LastModified           Size  ETag
obs://bucket-test2/123            2022-03-29T09:17:51Z  0B
"d41d8cd98f00b204e9800998ecf8427e"

obs://bucket-test2/1_2-3.txt      2022-03-29T09:17:51Z  0B
"d41d8cd98f00b204e9800998ecf8427e"

obs://bucket-test2/1_2-3_33.txt   2022-03-29T09:17:51Z  200B
"dcf204c11d791255adc63e61763c2426"

obs://bucket-test2/新建文本文档.txt 2022-03-29T09:17:51Z  0B
"d41d8cd98f00b204e9800998ecf8427e"

Total size of bucket: 200B
Folder number: 0
File number: 4
```

- 示例3：以Windows操作系统为例，运行**obsutil ls obs://bucket-test2/prefix**，查询桶内以prefix为前缀的对象。

```
obsutil ls obs://bucket-test2/prefix
Start at 2024-09-30 08:24:36.7057148 +0000 UTC

Listing objects .

Object list:
key                               LastModified           Size   ETag
obs://bucket-test2/prefix         2022-03-29T09:17:51Z  0B    "d41d8cd98f00b204e9800998ecf8427e"
obs://bucket-test2/prefix2        2022-03-29T09:17:51Z  0B    "d41d8cd98f00b204e9800998ecf8427e"

Total size of prefix [prefix]: 0B
Folder number: 0
File number: 2
```

- 更多运行示例，请参见。

## 参数说明

参数	约束	描述
bucket	必选	桶名。
prefix	可选	列举对象时的对象名前缀。 <b>说明</b> 如果为空，则表示列举桶内所有对象。
s	附加参数，可选	以精简格式显示查询结果。 <b>说明</b> 在精简格式下，返回结果只包含对象名。
d	附加参数，可选	仅列举当前目录下的对象和子目录，而非递归列举所有对象和子目录。 对于并行文件系统目录，大数据场景下（目录层级深、目录下文件多）的列举，建议使用此参数限制列举范围。 <b>说明</b> OBS中约定以“/”作为目录分隔符。
v	附加参数，可选	列举桶内多版本对象，列举结果包含最新版本的对象和历史版本的对象（如果有）。
marker	附加参数，可选	列举桶内对象的起始位置，返回结果是对象名按照字典序排序后该参数以后的所有对象。 <b>说明</b> 请参考理解该参数的使用方式。
versionIdMarker	附加参数，可选， 必须与v和marker配合使用	列举桶内多版本对象的起始位置，返回结果是对象名和版本号按照字典序排序后该参数以后的所有对象。 <b>说明</b> 如果versionIdMarker不是marker的一个版本号，则该参数无效。

参数	约束	描述
bf	附加参数, 可选	列举结果中字节数的显示格式。支持的值: <ul style="list-style-type: none"><li>• human-readable</li><li>• raw</li></ul> <b>说明</b> 如果未设置该参数, 则列举结果中字节数的显示格式由配置文件中的humanReadableFormat参数决定。
du	附加参数, 可选	快速返回所列举对象的总大小, 不显示具体的对象信息, 可与其他参数配合使用。 <b>说明</b> 如果所列举对象过多, 需稍候。 <b>注意</b> <ul style="list-style-type: none"><li>• Util 5.4.6 版本及以上支持该参数。</li></ul>
limit	附加参数, 可选	列举结果的最大个数, 小于等于0表示列举所有结果, 不设置时的默认值为1000。 <b>说明</b> 如果桶内对象较多, 建议设置该值限制每次列举的对象个数, 列举结束后, 如果仍有对象, 会在结果中返回下次请求的marker和versionIdMarker, 用于继续列举剩余的对象。
format	附加参数, 可选	指定以自定义格式打印列举结果。当前仅支持值[default], 指定列举结果在一行显示。
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

字段名	描述
Key	对象名。
LastModified	对象的最近一次修改时间。
Size	对象的大小。
ETag	对象的ETag值。对象的base64编码的128位MD5摘要。ETag是对象内容的唯一标识, 可以通过该值识别对象内容是否有变化。比如上传对象时ETag为A, 下载对象时ETag为B, 则说明对象内容发生了变化。ETag只反映变化的内容, 而不是其元数据。上传的对象或拷贝操作创建的对象, 都有唯一的ETag。



## 5.6 复制对象

### 功能说明

复制对象或按对象名前缀批量复制对象。

#### 须知

- 复制对象或批量复制时，请确保OBS桶中的源对象不发生变化，否则可能产生复制失败或者数据不一致的问题。
- 复制时需要要有被复制对象的读权限和目标桶的写权限。
- 如果没有开启客户端跨区域复制模式，复制的源桶和目标桶必须在同一个区域（Region）。
- 如果源桶是并行文件系统（POSIX），则目标桶不能是非并行文件系统。

### 命令结构

- Windows操作系统
  - 复制单个对象  
`obsutil cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun][-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]`
  - 批量复制对象  
`obsutil cp obs://srcbucket[/key] obs://dstbucket/[dest] -r [-dryRun][-f] [-flat] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]`
- macOS/Linux操作系统
  - 复制单个对象  
`./obsutil cp obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-crr] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]`
  - 批量复制对象  
`./obsutil cp obs://srcbucket[/key] obs://dstbucket/[dest] -r [-dryRun] [-f] [-flat] [-u] [-crr] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=*.xxx] [-exclude=*.xxx] [-timeRange=time1-time2] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]`

#### 说明

- 复制的源路径和目标路径不能完全相同。
- 批量复制时，源路径和目标路径不能相互嵌套（目标路径以源路径为前缀时，可能会产生递归复制；源路径以目标路径为前缀时，可能会产生覆盖拷贝）。

### 运行示例

- 以Windows操作系统为例，运行**obsutil cp obs://bucket-test/key obs://bucket-test2**，复制单个对象。

```
obsutil cp obs://bucket-test/key obs://bucket-test2
Start at 2024-09-30 08:30:09.0815415 +0000 UTC
```

```
Parallel: 3      Jobs: 3
Threshold: 50.00MB  PartSize: auto
CheckpointDir: xxxx

[=====] 100.00% 6/s 0s
Waiting for the copied key to be completed on server side.
Copy successfully, 19B, obs://bucket-test/key --> obs://bucket-test2/key
ext.txt, cost [1708], status [200], request id [00000192420D227E4017336A12F1DC22]
```

- 以Windows操作系统为例，运行**obsutil cp obs://bucket-test/temp/ obs://bucket-test2 -f -r**，批量复制对象。

```
obsutil cp obs://bucket-test/temp/ obs://bucket-test2 -r -f
Start at 2024-09-30 08:34:02.7819703 +0000 UTC
```

```
Parallel: 5      Jobs: 5
Threshold: 50.00MB  PartSize: auto
CheckpointDir: xxxx

Task id: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
OutputDir: xxxx
```

```
[=====] 100.00% 10/s 0s
Succeed count: 5      Failed count: 0
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:9.71, transferred size:
7.20MB]
Task id: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
```

- 更多运行示例，请参见。

## 参数说明

参数	约束	描述
srcbucket	必选	源桶名。
dstbucket	必选	目标桶名。
dest	可选	复制对象时的目标对象名，或批量复制时目标对象名前缀。

参数	约束	描述
key	复制对象时 必选 批量复制时 可选	<p>复制对象时的源对象名，或批量复制时源对象名前缀。</p> <p>规则如下：</p> <ul style="list-style-type: none"> <li>复制对象时，该值不能为空；如果dest为空，则将源对象复制到目标桶的根目录；如果dest以“/”结尾，则目标对象名为dest+源对象名；否则，目标对象名为dest；</li> <li>批量复制时，如果该值为空，则复制源桶中的所有对象；否则，复制源桶中以该值为对象名前缀的所有对象。目标对象名的确认规则为： <ul style="list-style-type: none"> <li>如果dest以“/”结尾，目标对象名为dest+源对象名；</li> <li>如果dest不以“/”结尾，目标对象名为dest+源对象名。</li> </ul> </li> </ul> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>批量复制且该值不为空时，如果未设置flat选项，则复制时源对象名会包含上一级父对象名前缀；如果设置了flat选项，则复制时源对象名不包含上一级父对象名前缀。</li> <li>请参考理解该参数的使用方式。</li> </ul>
fr	附加参数， 复制对象时 可选	复制对象时生成结果清单文件。
flat	附加参数， 批量复制时 可选	批量复制时，不包含上一级父对象名前缀。
dryRun	附加参数， 可选	测试模式运行，不执行实际的复制操作。
crr	附加参数， 可选	<p>复制时使用客户端跨区域复制模式，以通过数据流的方式从源桶直接复制数据到目标桶，且两个桶可以是任意两个OBS服务的桶。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>如果设置了该参数，必须确保更新了配置文件中客户端跨区域复制的相关配置信息，具体可参考。</li> <li>复制时源桶对应的配置信息为配置文件中的：akCrr/skCrr/tokenCrr/endpointCrr，目标桶对应的配置信息为配置文件中的：ak/sk/token/endpoint。</li> </ul> <p><b>须知</b></p> <p>开启跨区域复制后，因为会占用执行命令主机上传和下载的双向带宽、CPU和内存等资源，可能会影响执行命令主机的性能。</p>
vlenth	附加参数， 可选	<p>复制完成后，验证目标桶对象大小是否与源桶中对象的大小一致。</p> <p><b>说明</b></p> <p>必须与crr参数配合使用。</p>

参数	约束	描述
vmd5	附加参数, 可选	复制完成后, 验证目标桶的MD5值是否与源桶中对象的MD5值一致。 <b>说明</b> <ul style="list-style-type: none"><li>• 必须与crr参数配合使用。</li><li>• 源桶中对象必须包含元数据x-obs-meta-md5chksum, 否则会忽略MD5值校验。MD5值校验通过后, 会将该值设置为目标对象元数据x-obs-meta-md5chksum, 用于下载或复制时校验MD5。</li></ul> <b>注意</b> <p>如果您的业务涉及加密场景, 请勿使用该参数。</p>
u	附加参数, 可选	增量复制操作, 设置该参数后, 复制每个源对象时会对比目标桶中对应路径的目标对象, 仅在目标对象不存在, 或者目标对象大小与源对象大小不一致, 或者目标对象的最后修改时间早于源对象的最后修改时间时进行复制。
p	附加参数, 可选	每个分段复制任务的最大并发数, 默认为配置文件中的defaultParallels。
threshold	附加参数, 可选	开启分段复制任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"><li>• 如果待复制的源对象小于该阈值, 则使用直接复制模式; 否则使用分段复制模式。</li><li>• 直接复制模式不会产生断点记录文件, 不支持断点续传。</li><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
versionId	附加参数, 复制对象时 可选	复制单个对象时可指定的源对象版本号。
acl	附加参数, 可选	复制对象时可指定的目标对象的预定义访问策略。支持的值: <ul style="list-style-type: none"><li>• private</li><li>• public-read</li><li>• public-read-write</li><li>• bucket-owner-full-control</li></ul> <b>说明</b> <p>以上四个值分别对应: 私有读写、公共读、公共读写、桶拥有者完全控制, 四种预定义访问策略。</p>

参数	约束	描述
meta	附加参数, 可选	复制对象时可指定的目标对象的标准和自定义元数据。格式为: key1:value1#key2:value2#key3:value3。 <b>说明</b> <ol style="list-style-type: none"><li>以上值表示复制对象完成后, 桶内目标对象包含三组自定义元数据, 分别为: key1:value1、key2:value2、key3:value3。</li><li>标准元数据头包括: Content-Type、Content-Encoding、Cache-Control、Content-Disposition、Content-Language、Expires。</li></ol>
ps	附加参数, 可选	每个分段复制任务的段大小, 单位: 字节, 取值范围是100KB~5GB, 默认为配置文件中的defaultPartSize。 <b>说明</b> <ul style="list-style-type: none"><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>支持配置为auto, 此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。 <b>说明</b> <p>每个分段复制任务会产生唯一对应的断点记录文件并保存至该文件夹的copy子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
r	附加参数, 批量复制时 必选	按指定源桶的对象名前缀批量复制对象。
f	附加参数, 批量复制时 可选	强制操作, 不进行询问提示。
j	附加参数, 批量复制时 可选	批量复制时的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> <p>工具会保证该值至少为1。</p>

参数	约束	描述
exclude	附加参数, 批量复制时 可选	<p>不包含源对象的匹配模式, 如: *.txt。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li><li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li><li>如果待复制的对象名匹配该参数, 则跳过该对象的复制。</li></ul> <p><b>须知</b></p> <ul style="list-style-type: none"><li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li><li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li><li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li><li>支持指定多个exclude参数, 如 -exclude=*.xxx -exclude=*.xxx。</li></ul>
include	附加参数, 批量复制时 可选	<p>包含源对象的匹配模式, 如: *.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符。</li><li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li><li>首先执行exclude的匹配规则, 如果待复制的对象名不匹配exclude, 则判断待复制的对象名是否匹配该参数, 如果匹配则复制该对象, 否则跳过该文件的复制。</li></ul> <p><b>须知</b></p> <ul style="list-style-type: none"><li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li><li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li><li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li><li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx。</li></ul>

参数	约束	描述
timeRange	附加参数, 批量复制时 可选	<p>复制对象时的时间段匹配模式, 仅复制最后修改时间在该时间段内的对象。</p> <p>该匹配模式优先级低于对象名匹配模式: exclude和include, 优先执行对象名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>使用“time1-time2”, 代表匹配的时间段, 其中time1必须小于等于time2, 格式为yyyyMMddHHmmss。</li><li>时间格式支持自动补全, 例如yyyyMMdd等价于yyyyMMdd000000, yyyyMM等价于yyyyMM01000000。</li><li>使用“*-time2”, 代表匹配最后修改时间在time2之前的所有文件, 使用“time1-*”, 代表匹配最后修改时间time1之后的所有文件。</li></ul> <p><b>须知</b></p> <ul style="list-style-type: none"><li>该匹配模式表示的时间是UTC时间。</li><li>该匹配模式仅适用于对象名非“/”结尾的对象。</li></ul>
mf	附加参数, 可选	设置名称匹配模式(include、exclude)和时间匹配模式(timeRange)对对象名以“/”结尾的对象也生效。
o	附加参数, 可选	<p>生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件(可能包含成功结果、失败结果和警告结果三个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>结果清单文件命名规则: cp_{succeed   failed   warning}_report_时间_TaskId.txt。</li><li>单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件“cp_failed_report_时间_TaskId.txt”及日志路径下的。</li></ul>
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.7 移动对象

### 功能说明

移动对象或按对象名前缀批量移动对象。

### 须知

- 移动对象时，请确保OBS桶中的源对象不发生变化，否则可能产生移动失败或者数据不一致的问题。
- 移动对象或批量移动成功后，源端对象会被删除。
- 如果源桶是并行文件系统（POSIX），则目标桶不能是对象存储桶。

## 命令结构

- Windows操作系统

- 移动单个对象

```
obsutil mv obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]
```

- 批量移动对象

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

- macOS/Linux操作系统

- 移动单个对象

```
./obsutil mv obs://srcbucket/key obs://dstbucket/[dest] [-dryRun] [-u] [-p=1] [-threshold=52428800] [-versionId=xxx] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-cpd=xxx] [-fr] [-o=xxx] [-config=xxx]
```

- 批量移动对象

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

### 说明

- 移动的源路径和目标路径不能完全相同。
- 批量移动时，源路径和目标路径不能相互嵌套。
- 不支持对并行文件系统的批量移动的操作。

## 运行示例

- 以Windows操作系统为例，运行**obsutil mv obs://bucket-test/key obs://bucket-test2**，移动单个对象。

```
obsutil mv obs://bucket-test/key obs://bucket-test2
Start at 2024-09-30 08:36:01.3934921 +0000 UTC
```

```
Parallel: 5          Jobs: 5
Threshold: 50.00MB  PartSize: auto
CheckpointDir: xxxx
```

```
[=====] 100.00% 6/s 0s
Waiting for the copied key to be completed on server side.
```

```
Move successfully, 19B, obs://bucket-test/key --> obs://bucket-test2/key, cost [1815], status [200], request id [00000192421282AC401423183A8B83A1]
```

- 以Windows操作系统为例，运行**obsutil mv obs://bucket-test/temp/ obs://bucket-test2 -f -r**，批量移动对象。

```
obsutil mv obs://bucket-test/temp/ obs://bucket-test2 -f -r
Start at 2024-09-30 08:37:32.2454905 +0000 UTC
```

```
Parallel: 5          Jobs: 5
Threshold: 50.00MB  PartSize: auto
```



```
CheckpointDir: xxxx
Task id: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
OutputDir: xxxx

[=====] 100.00% 10/s 0s
Succeed count: 5    Failed count: 0
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:9.71, transferred size:
7.20MB]
Task id: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
```

## 参数说明

参数	约束	描述
srcbucket	必选	源桶名。
dstbucket	必选	目标桶名。
dest	可选	移动对象时的目标对象名，或批量移动时目标对象名前缀。
key	移动对象时 必选 批量移动时 可选	<p>移动对象时的源对象名，或批量移动时源对象名前缀。</p> <p>规则如下：</p> <ul style="list-style-type: none"> <li>移动对象时，该值不能为空；如果dest为空，则将源对象移动到目标桶的根目录；如果dest以“/”结尾，则目标对象名为dest+源对象名；否则，目标对象名为dest；</li> <li>批量移动时，如果该值为空，则移动源桶中的所有对象；否则，移动源桶中以该值为对象名前缀的所有对象。目标对象名的确认规则为： <ul style="list-style-type: none"> <li>如果dest以“/”结尾，目标对象名为dest+源对象名；</li> <li>如果dest不以“/”结尾，目标对象名为dest+源对象名。</li> </ul> </li> </ul> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>批量移动且该值不为空时，如果未设置flat选项，则移动时源对象名会包含上一级父对象名前缀；如果设置了flat选项，则移动时源对象名不包含上一级父对象名前缀。</li> <li>请参考移动<a href="#">命令结构</a>理解该参数的使用方式。</li> </ul>
fr	附加参数， 移动对象时 可选	移动对象时生成结果清单文件。
flat	附加参数， 批量移动时 可选	移动时，不包含上一级父对象名前缀。
dryRun	附加参数， 可选	测试模式运行，不执行实际的移动操作。

参数	约束	描述
u	附加参数, 可选	增量移动操作, 设置该参数后, 移动每个源对象时会对比目标桶中对应路径的目标对象, 仅在目标对象不存在, 或者目标对象大小与源对象大小不一致, 或者目标对象的最后修改时间早于源对象的最后修改时间时进行移动。 <b>说明</b> 如果目标对象与源对象大小以及修改时间都一致, 此时会直接删除源对象, 而不进行移动操作。
p	附加参数, 可选	每个分段移动任务的最大并发数, 默认为配置文件中的defaultParallels。
threshold	附加参数, 可选	开启分段移动任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"><li>• 如果待移动的源对象小于该阈值, 则使用直接移动模式; 否则使用分段移动模式。</li><li>• 直接移动模式不会产生断点记录文件, 不支持断点续传。</li><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
versionId	附加参数, 移动对象时 可选	移动单个对象时可指定的源对象版本号。 <b>说明</b> POSIX桶无version概念, 故POSIX桶不支持versionId选项。
acl	附加参数, 可选	移动对象时可指定的目标对象的预定义访问策略。支持的值: <ul style="list-style-type: none"><li>• private</li><li>• public-read</li><li>• public-read-write</li><li>• bucket-owner-full-control</li></ul> <b>说明</b> 以上四个值分别对应: 私有读写、公共读、公共读写、桶拥有者完全控制, 四种预定义访问策略。
meta	附加参数, 可选	移动对象时可指定的目标对象的自定义元数据。格式为: key1:value1#key2:value2#key3:value3。 <b>说明</b> 以上值表示移动对象完成后, 桶内目标对象包含三组自定义元数据, 分别为: key1:value1、key2:value2、key3:value3。

参数	约束	描述
ps	附加参数, 可选	每个分段移动任务的段大小, 单位: 字节, 取值范围是100KB~5GB, 默认为配置文件中的defaultPartSize。 <b>说明</b> <ul style="list-style-type: none"><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>支持配置为auto, 此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。 <b>说明</b> <p>每个分段移动任务会产生唯一对应的断点记录文件并保存至该文件夹的copy子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
r	附加参数, 批量移动时 必选	按指定源桶的对象名前缀批量移动对象。
f	附加参数, 批量移动时 可选	强制操作, 不进行询问提示。
j	附加参数, 批量移动时 可选	批量移动时的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> <p>工具会保证该值至少为1。</p>
exclude	附加参数, 批量移动时 可选	不包含源对象的匹配模式, 如: *.txt。 <b>说明</b> <ul style="list-style-type: none"><li>支持“*”匹配多个任意字符和“?”匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li><li>您可以使用“\*”代表匹配“*”字符本身, 使用“\?”代表匹配“?”字符本身。</li><li>如果待移动的对象名匹配该参数, 则跳过该对象的移动。</li></ul> <b>须知</b> <ul style="list-style-type: none"><li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li><li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li><li>该匹配模式仅适用于对象名非“/”结尾的对象。</li><li>支持指定多个exclude参数, 如-exclude=*.xxx -exclude=*.xxx</li></ul>

参数	约束	描述
include	附加参数, 批量移动时 可选	<p>包含源对象的匹配模式, 如: *.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符。</li><li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li><li>首先执行exclude的匹配规则, 如果待移动的对象名不匹配exclude, 则判断待移动的对象名是否匹配该参数, 如果匹配则移动该对象, 否则跳过该文件的复制。</li></ul> <p><b>须知</b></p> <ul style="list-style-type: none"><li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li><li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li><li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li><li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx</li></ul>
timeRange	附加参数, 批量移动时 可选	<p>移动对象时的时间段匹配模式, 仅移动最后修改时间在该时间段内的对象。</p> <p>该匹配模式优先级低于对象名匹配模式: exclude和include, 优先执行对象名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>使用 “time1-time2”, 代表匹配的时间段, 其中time1必须小于等于time2, 格式为yyyyMMddHHmmss。</li><li>时间格式支持自动补全, 例如yyyyMMdd等价于yyyyMMdd000000, yyyyMM等价于yyyyMM01000000。</li><li>使用 “*-time2”, 代表匹配最后修改时间在time2之前的所有文件, 使用 “time1-*”, 代表匹配最后修改时间time1之后的所有文件。</li></ul> <p><b>须知</b></p> <ul style="list-style-type: none"><li>该匹配模式表示的时间是UTC时间。</li><li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li></ul>
mf	附加参数, 可选	<p>设置名称匹配模式 ( include、exclude ) 和时间匹配模式 ( timeRange ) 对对象名以 “/” 结尾的对象也生效。</p>

参数	约束	描述
o	附加参数, 可选	生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件(可能包含成功结果、失败结果和警告结果三个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"><li>结果清单文件命名规则: mv_{succeed   failed   warning}_report_时间_TaskId.txt。</li><li>单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件"mv_failed_report_时间_TaskId.txt"及日志路径下的<a href="#">日志文件</a>;</li></ul>
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.8 下载对象

### 功能说明

下载对象或按对象名前缀批量下载对象到本地。

#### 须知

- 下载对象或批量下载时, 请确保OBS桶中的源对象不发生变化, 否则可能产生下载失败或者数据不一致的问题。

### 命令结构

- Windows操作系统

- 下载单个对象

```
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]
```

- 批量下载对象

```
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]
```

- macOS/Linux操作系统

- 下载单个对象

```
./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]
```

- 批量下载对象

```
./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]
```

## 运行示例

- 以Windows操作系统为例，运行**obsutil cp obs://bucket-test/key d:\temp\test.txt**，下载单个对象。

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

```
Start at 2024-09-30 08:39:34.180766 +0000 UTC
```

```
Parallel: 5          Jobs: 5
Threshold: 50.00MB  PartSize: auto
VerifyLength: false VerifyMd5: false
CheckpointDir: xxxx
TempFileDir: xxxx
```

```
[=====] 100.00% 4.86 KB/s 8.46MB/8.46MB 0s
Download successfully, 8.46MB, obs://bucket-test/key --> d:\temp\test.txt, cost [41], status [200], request id [000001924215BEC84019EDF4044A5451]
```

- 以Windows操作系统为例，运行**obsutil cp obs://bucket-test/temp d:\ -f -r**，批量下载对象。

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

```
Start at 2024-09-30 08:41:56.0306522 +0000 UTC
```

```
Parallel: 5          Jobs: 5
Threshold: 50.00MB  PartSize: auto
VerifyLength: false VerifyMd5: false
CheckpointDir: xxxx
```

```
Task id: 3066a4b0-4d21-4929-bb84-4829c32cbd0f
OutputDir: xxxx
TempFileDir: xxxx
```

```
[=====] 100.00% tps:17.86 155.59 KB/s
7.20MB/7.20MB 0s
Succeed count: 6      Failed count: 0
Succeed bytes: 70B
Metrics [max cost:153 ms, min cost:129 ms, average cost:92.00 ms, average tps:70, transferred size: 7.20MB]
```

```
Task id: 3066a4b0-4d21-4929-bb84-4829c32cbd0f
```

- 更多运行示例，请参见。

## 参数说明

参数	约束	描述
file_or_folder_url	下载对象时必选	本地文件/文件夹路径。
folder_url	批量下载时必选	本地文件夹路径。
bucket	必选	桶名。

参数	约束	描述
key	下载对象时必选 批量下载时可选	<p>待下载的对象名, 或批量下载的对象名前缀。</p> <p>下载对象时, 该值不能为空, 最终保存在本地的位置及命名规则如下:</p> <ul style="list-style-type: none"> <li>• 如果file_or_folder_url不存在, 则工具会判断其是否以“/”或“\”结尾, 如果是, 则会以该路径创建文件夹, 将对象下载到该新建的文件夹下;</li> <li>• 如果file_or_folder_url不存在, 且不以“/”或“\”结尾, 则以该值作为文件名, 将对象下载到本地;</li> <li>• 如果file_or_folder_url为已存在的文件, 则以该值作为文件名, 将对象覆盖下载到本地;</li> <li>• 如果file_or_folder_url为已存在的文件夹, 则以对象名为文件名, 将对象覆盖下载到file_or_folder_url指定的文件夹下。</li> </ul> <p>批量下载时, 最终保存在本地的位置规则如下:</p> <ul style="list-style-type: none"> <li>• 如果该值为空, 则将整个桶的内容下载到folder_url指定的文件夹下;</li> <li>• 如果该值不为空, 则将桶内以该值作为对象名前缀的内容下载到folder_url指定的文件夹下。</li> </ul> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>• 批量下载且该值不为空时, 如果未设置flat选项, 则下载的本地文件会包含上一级父对象名前缀; 如果设置了flat选项, 则下载的本地文件不包含上一级父对象名前缀。</li> <li>• 请参考理解该参数的使用方式。</li> </ul>
fr	附加参数, 下载对象时可选	下载对象时生成结果清单文件。
flat	附加参数, 批量下载时可选	批量下载时, 不包含上一级父对象名前缀。
tempFileDir	附加参数, 可选	<p>分段下载时保存临时文件的文件夹, 默认为配置文件中的defaultTempFileDir。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>• 如果该值为空, 且配置文件中的defaultTempFileDir也为空, 则分段下载时生成的临时文件会保存在待下载文件的同目录下并以.obs.temp结尾。</li> <li>• 由于分段下载时生成的临时文件会保存在该目录, 请确保运行obsutil的用户对该路径有写权限。</li> <li>• 该路径的可用空间需要大于待下载对象的大小。</li> </ul>
dryRun	附加参数, 可选	测试模式运行, 不执行实际的下载操作。

参数	约束	描述
u	附加参数, 可选	增量下载操作, 设置该参数后, 下载每个对象时会对比本地对应路径的文件, 仅在文件不存在, 或者文件大小与对象大小不一致, 或者文件的最后修改时间早于对象的最后修改时间时进行下载。
vlength	附加参数, 可选	下载完成后, 验证本地文件大小是否与桶中对象的大小一致。
vmd5	附加参数, 可选	下载完成后, 验证本地文件的MD5值是否与桶中对象的MD5值一致。 <b>说明</b> 桶中对象必须包含元数据x-obs-meta-md5chksum, 否则会忽略MD5值校验。 <b>注意</b> 如果您的业务涉及加密场景, 请勿使用该参数。
p	附加参数, 可选	每个分段下载任务的最大并发数, 默认为配置文件中的defaultParallels。
threshold	附加参数, 可选	开启分段下载任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"><li>• 如果待下载的对象小于该阈值, 则使用直接下载模式; 否则使用分段下载模式。</li><li>• 直接下载模式不会产生断点记录文件, 不支持断点续传。</li><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
versionId	附加参数, 下载对象时可选	下载单个对象时可指定的对象版本号。
ps	附加参数, 可选	每个分段下载任务的段大小, 单位: 字节, 默认为配置文件中的defaultPartSize。 <b>说明</b> <ul style="list-style-type: none"><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>• 支持配置为auto, 此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。 <b>说明</b> 每个分段下载任务会产生唯一对应的断点记录文件并保存至该文件夹的down子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。
r	附加参数, 批量下载时必选	按指定的对象名前缀批量下载对象。



参数	约束	描述
f	附加参数, 批量下载时可选	强制操作, 不进行询问提示。
j	附加参数, 批量下载时可选	批量下载时的最大并发数, 默认为配置文件中的 defaultJobs。 <b>说明</b> 工具会保证该值至少为1。
exclude	附加参数, 批量下载时可选	不包含源对象的匹配模式, 如: *.txt。 <b>说明</b> <ul style="list-style-type: none"><li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li><li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li><li>如果待下载的对象名匹配该参数, 则跳过该对象的复制。</li></ul> <b>须知</b> <ul style="list-style-type: none"><li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li><li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li><li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li><li>支持指定多个exclude参数, 如-exclude=*.xxx -exclude=*.xxx</li></ul>

参数	约束	描述
include	附加参数, 批量下载时可选	<p>include参数规定了下载对象的对象名格式, 满足该格式的对象才会被下载。例如 “*.jpg” 规定只有对象名后缀为 “.jpg” 的对象会被下载。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符。</li> <li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li> <li>首先执行exclude的匹配规则, 如果待下载的对象名不匹配exclude, 则判断待下载的对象名是否匹配该参数, 如果匹配则下载该对象, 否则跳过该文件的下载。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> <li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx</li> </ul>
timeRange	附加参数, 批量下载时可选	<p>下载对象时的时间段匹配模式, 仅下载最后修改时间在该时间段内的对象。</p> <p>该匹配模式优先级低于对象名匹配模式: exclude和include, 优先执行对象名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>使用 “time1-time2”, 代表匹配的时间段, 其中time1必须小于等于time2, 格式为yyyyMMddHHmmss。</li> <li>时间格式支持自动补全, 例如yyyyMMdd等价于yyyyMMdd000000, yyyyMM等价于yyyyMM01000000。</li> <li>使用 “*-time2”, 代表匹配最后修改时间在time2之前的所有文件, 使用 “time1-*”, 代表匹配最后修改时间time1之后的所有文件。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>该匹配模式表示的时间是UTC时间。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> </ul>
mf	附加参数, 可选	<p>设置名称匹配模式 ( include、exclude ) 和时间匹配模式 ( timeRange ) 对对象名以 “/” 结尾的对象也生效。</p>

参数	约束	描述
o	附加参数, 可选	生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件(可能包含成功结果、失败结果和警告结果三个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"><li>结果清单文件命名规则: cp_{succeed   failed   warning}_report_时间_TaskId.txt。</li><li>单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件"cp_failed_report_时间_TaskId.txt"及日志路径下的。</li></ul>
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.9 生成对象的下载链接

### 功能说明

生成指定桶内对象的下载链接, 或按对象名前缀批量生成桶内对象的下载链接。

### 命令结构

- Windows操作系统
  - 生成单个对象的下载链接  
`obsutil sign obs://bucket/key [-e=300] [-config=xxx]`
  - 按对象名前缀批量生成对象的下载链接  
`obsutil sign obs://bucket[/key] -r [-e=300] [-timeRange=time1-time2] [-include=*xxx] [-exclude=*xxx] [-o=xxx] [-config=xxx]`
- macOS/Linux操作系统
  - 生成单个对象的下载链接  
`./obsutil sign obs://bucket/key [-e=300] [-config=xxx]`
  - 按对象名前缀批量生成对象的下载链接  
`./obsutil sign obs://bucket[/key] -r [-e=300] [-timeRange=time1-time2] [-include=*xxx] [-exclude=*xxx] [-o=xxx] [-config=xxx]`

### 运行示例

- 以Windows操作系统为例, 运行**obsutil sign obs://bucket-test/test.txt**, 生成单个对象下载链接。

```
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

- 以Windows操作系统为例，运行**obsutil sign obs://bucket-test/test/ -r**，批量生成对象下载链接。

```
obsutil sign obs://bucket-test/test/ -r
```

Generate download urls for objects .

Generate the download url(s) for the objects in the bucket [**bucket-test**] finished  
Task id: af4dc692-6a88-4541-8156-ff1a889d2288

### 📖 说明

- 由于可能存在对象数量庞大的场景，OBS Util会将对象的下载链接归档到结果清单文件目录下具体的文件，该文件名与Task Id相关联，比如上例的Task Id为af4dc692-6a88-4541-8156-ff1a889d2288，那么存在sign\_succeed\_report\_{时间戳}\_af4dc692-6a88-4541-8156-ff1a889d2288.txt文件。
- 结果清单文件目录默认为用户目录的子文件夹.obsutil\_output，也可以通过-o参数指定生成目录。

## 参数说明

参数	约束	描述
bucket	必选	桶名。
key	可选	生成单个对象下载链接的对象名，或批量生成对象下载链接的对象名前缀。
e	附加参数，可选	生成的对象下载链接的过期时间（单位：秒），最小值为60秒，默认为300秒。
r	附加参数，批量生成时必选	按指定的对象名前缀批量生成对象的下载链接。
exclude	附加参数，批量生成时可选	<p>不包含对象的匹配模式，如：*.txt。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持“*”匹配多个任意字符和“?”匹配单个任意字符，例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li> <li>您可以使用“\*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。</li> <li>如果待下载的对象名匹配该参数，则跳过该对象的复制。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。</li> <li>该匹配模式仅适用于对象名非“/”结尾的对象。</li> <li>支持指定多个exclude参数，如-exclude=*.xxx -exclude=*.xxx</li> </ul>

参数	约束	描述
include	附加参数，批量生成时可选	<p>包含对象的匹配模式，如：*.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持“*”匹配多个任意字符和“?”匹配单个任意字符。</li> <li>您可以使用“\*”代表匹配“*”字符本身，使用“\?”代表匹配“?”字符本身。</li> <li>首先执行exclude的匹配规则，如果待下载的对象名不匹配exclude，则判断待下载的对象名是否匹配该参数，如果匹配则下载该对象，否则跳过该文件的下载。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式（macOS/Linux操作系统使用单引号，Windows操作系统使用双引号）防止特殊符号被操作系统转义，导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径（含从根路径开始的对象前缀和对象名，例如，桶内对象路径为obs://bucket/src1/src2/test.txt，则对象的全路径为src1/src2/test.txt）。</li> <li>该匹配模式仅适用于对象名非“/”结尾的对象。</li> <li>支持指定多个include参数，如-include=*.xxx -include=*.xxx</li> </ul>
timeRange	附加参数，可选	<p>生成对象下载链接时的时间段匹配模式，仅生成最后修改时间在该时间段内对象的链接。</p> <p>该匹配模式优先级低于对象名匹配模式：exclude和include，优先执行对象名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>该匹配模式表示的时间是UTC时间。</li> <li>该匹配模式仅适用于对象名非“/”结尾的对象。</li> <li>使用“time1-time2”，代表匹配的时间段，其中time1必须小于等于time2，格式为yyyyMMddHHmmss。</li> <li>时间格式支持自动补全，例如yyyyMMdd等价于yyyyMMdd000000，yyyyMM等价于yyyyMM01000000。</li> <li>使用“*-time2”，代表匹配最后修改时间在time2之前的所有文件，使用“time1-*”，代表匹配最后修改时间time1之后的所有文件。</li> </ul>
o	附加参数，批量生成时可选	<p>生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果和失败结果两个文件），默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>结果清单文件命名规则：sign_{succeed   failed}_report_时间_TaskId.txt。 单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li> </ul>
config	附加参数，可选	<p>运行当前命令时的自定义配置文件，可支持的配置参数请参考。</p>

## 5.10 删除对象

### 功能说明

- 删除指定的对象。
- 按指定的对象名前缀批量删除对象。



#### 警告

删除对象后无法恢复，请谨慎使用。

### 使用建议

对于并行文件系统目录，大数据场景下（目录层级深、目录下文件多）的删除，可能会因超时而删除失败，建议使用：

1. hadoop 客户端（嵌套OBS客户端插件OBSA）删除目录：`hadoop fs -rmdir obs://{并行文件系统名}/{目录名}`
2. 给目录配置生命周期规则，通过生命周期后台删除。

### 命令结构

- Windows操作系统
  - 删除单个对象  
`obsutil rm obs://bucket/key [-f] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]`
  - 批量删除对象  
`obsutil rm obs://bucket/[key] -r [-j=1] [-f] [-v] [-o=xxx] [-config=xxx]`
- macOS/Linux操作系统
  - 删除单个对象  
`./obsutil rm obs://bucket/key [-f] [-versionId=xxx] [-fr] [-o=xxx] [-config=xxx]`
  - 批量删除对象  
`./obsutil rm obs://bucket/[key] -r [-j=1] [-f] [-v] [-o=xxx] [-config=xxx]`

### 运行示例

- 以Windows操作系统为例，运行**`obsutil rm obs://bucket-test/key -f`**，删除bucket-test桶中名为key的单个对象。

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

```
Start at 2024-09-25 04:48:10.1147483 +0000 UTC
```

```
Delete object [key] in the bucket [bucket-test] successfully, cost [152], request id [0000016979E1D2B2860BB5181229C72C]
```

- 以Windows操作系统为例，运行**`obsutil rm obs://bucket-test -r -f`**，删除bucket-test桶中所有对象。

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

```
Start at 2024-09-30 08:46:55.5335644 +0000 UTC
```

```
[=====] 100.00% 21s
```

```
Succeed count: 1313 Failed count: 0
```

```
Task id: 95936984-f81a-441a-bba0-1fd8254d9241
```

- 以Windows操作系统为例，运行`obsutil rm obs://bucket-test/key -r -f`，删除bucket-test桶中所有以key为前缀的对象和文件夹。

```
obsutil rm obs://bucket-test/key -r -f
Start at 2024-09-30 08:49:09.5602115 +0000 UTC
```

```
[=====] 100.00% 21s
Succeed count: 10   Failed count: 0
Task id: 79ab59ec-7e00-4f22-8c88-465faa834125
```

## 参数说明

参数	约束	描述
bucket	必选	桶名。
key	删除单个对象时必选 批量删除对象时可选	待删除的对象名，或批量删除的对象名前缀。 <b>说明</b> 批量删除对象时，如果该参数为空，则表示删除桶中所有的对象。
fr	附加参数，删除单个对象时可选	删除单个对象时生成结果清单文件。
f	附加参数，可选	强制操作，不进行询问提示。
versionId	附加参数，删除单个对象时可选	待删除对象的版本号。
r	附加参数，批量删除对象时必选	按指定的对象名前缀批量删除对象。 <b>注意</b> 批量删除对象时，会将与前缀匹配的所有对象删除，请谨慎使用。
j	附加参数，批量删除对象时可选	批量删除对象时的最大并发数，默认为配置文件中的defaultJobs。 <b>说明</b> 工具会保证该值至少为1。
v	附加参数，批量删除对象时可选	按指定的对象名前缀批量删除多版本对象和多版本删除标记。

参数	约束	描述
o	附加参数, 可选	生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件(可能包含成功结果和失败结果两个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"><li>结果清单文件命名规则: rm_{succeed   failed}_report_时间_TaskId.txt。</li><li>单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件"rm_failed_report_时间_TaskId.txt"及日志路径下的;</li></ul>
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.11 增量同步上传对象

### 功能说明

将本地源路径下的所有内容同步到OBS指定目标桶, 使两边内容保持一致。此处的增量同步有两层含义:

1. 增量, 依次比较源文件和目标对象, 只上传存在变化的源文件。
2. 同步, 命令执行完成后, 保证本地源路径是OBS指定目标桶的子集, 即本地源路径下的所有文件均能在OBS指定目标桶中找到对应对象。

#### 须知

- 同步上传时, 请确保本地文件或文件夹在同步过程不发生变化, 否则可能产生同步失败或者数据不一致的问题。
- 同步上传每个文件时会对比桶中对应路径的对象, 仅在对象不存在, 或者对象大小与文件大小不一致, 或者对象的最后修改时间早于文件的最后修改时间时进行上传。

### 命令结构

- Windows操作系统
  - 同步上传文件

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



- 同步上传文件夹  
obsutil sync folder\_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=\*.xxx] [-exclude=\*.xxx] [-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]
- macOS/Linux操作系统
  - 同步上传文件  
./obsutil sync file\_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-o=xxx] [-cpd=xxx] [-fr] [-config=xxx]
  - 同步上传文件夹  
./obsutil sync folder\_url obs://bucket[/key] [-arcDir=xxx] [-dryRun] [-link] [-vlength] [-vmd5] [-j=1] [-p=1] [-threshold=52428800] [-acl=xxx] [-meta=aaa:bbb#ccc:ddd] [-ps=auto] [-include=\*.xxx] [-exclude=\*.xxx] [-timeRange=time1-time2] [-at] [-mf] [-o=xxx] [-cpd=xxx] [-config=xxx]

## 运行示例

- 以Windows操作系统为例，运行**obsutil sync d:\temp\test.txt obs://bucket-test/key**，同步上传文件。  
**obsutil sync d:\temp\test.txt obs://bucket-test/key**  
Start at 2024-09-25 04:48:10.1147483 +0000 UTC  
Parallel: 5           Jobs: 5  
Threshold: 50.00MB    PartSize: auto  
VerifyLength: false    VerifyMd5: false  
CheckpointDir: C:\Users\Administrator\.obsutil\_checkpoint  
  
[=====] 100.00% 1.68 MB/s 8.46MB/  
8.46MB 5s  
Upload successfully, 8.46MB, d:\temp\test.txt --> obs://bucket-test/key, cost [55], status [200], request id [00000192421F4E224012B8470C0CCDC]
- 以Windows操作系统为例，运行**obsutil sync d:\temp obs://bucket-test/temp**，同步上传文件夹。  
**obsutil sync d:\temp obs://bucket-test/temp**  
Start at 2024-09-25 04:48:10.1147483 +0000 UTC  
Parallel: 5           Jobs: 5  
Threshold: 50.00MB    PartSize: auto  
VerifyLength: false    VerifyMd5: false  
CheckpointDir: C:\Users\Administrator\.obsutil\_checkpoint  
Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed  
  
OutputDir: C:\Users\Administrator\.obsutil\_output  
  
[=====] 100.00% tps:35.71 2.02 KB/s  
7.20MB/7.20MB 0s  
Succeed count: 5      Failed count: 0  
Succeed bytes: xxxx  
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71, transferred size: 7.20MB]  
Task id: 104786c8-27c2-48fc-bc6a-5886596fb0ed
- 更多运行示例，请参见[同步上传示例](#)。

## 参数说明

参数	约束	描述
file_url	同步上传文件时必选	本地文件路径。
folder_url	同步上传文件夹时必选	本地文件夹路径。
bucket	必选	桶名。
key	可选	同步上传文件时的对象名/对象名前缀, 或同步上传文件夹时的对象名前缀。 规则如下: <ul style="list-style-type: none"><li>同步上传文件时, 如果该值为空, 则将对象上传到桶的根目录, 对象名为文件名。如果该值不为空, 如果该值以“/”结尾, 则以该值作为对象名前缀上传对象, 对象名为该值+文件名; 否则, 将该值作为对象名进行上传;</li><li>同步上传文件夹时, 如果该值为空, 则将桶的根目录下的所有对象与本地文件夹下的所有内容保持一致; 如果该值不为空, 则将对象名前缀为该值的所有对象与本地文件夹下的所有内容保持一致。</li></ul> <b>说明</b> <ul style="list-style-type: none"><li>同步上传文件夹时, 如果该值不为空且不以“/”结尾, 则工具会自动为该值末尾加上“/”作为对象名前缀。</li><li>请参考<a href="#">同步上传示例</a>理解该参数的使用方式。</li></ul>
fr	附加参数, 同步上传文件时可选	同步上传文件时生成结果清单文件。
arcDir	附加参数, 可选	同步上传文件成功后的归档路径, 上传成功后的文件会移动到该路径下。
dryRun	附加参数, 可选	测试模式运行, 不执行实际的上传操作。
link	附加参数, 可选	上传软链接文件/文件夹指向的真实路径。 <b>须知</b> <ul style="list-style-type: none"><li>如果未指定该参数, 而待上传的文件是一个软链接且该软链接指向的目标文件不存在, 则Windows操作系统会产生“The system cannot find the file specified”的上传失败异常, 而macOS/Linux操作系统会产生“No such file or directory”的上传失败异常。</li><li>文件夹软链接不能形成环, 否则上传会以panic的形式退出; 如果不希望产生panic, 请在配置文件中配置panicForSymbolicLinkCircle为false。</li></ul>
vlenth	附加参数, 可选	同步上传完成后, 验证上传到桶中对象的大小是否与本地文件大小一致。

参数	约束	描述
vmd5	附加参数, 可选	<p>同步上传完成后, 验证上传到桶中对象的MD5值是否与本地文件的MD5值一致。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>如果待上传的本地文件较大, 使用该参数将会因为计算MD5而导致整体性能下降。</li><li>MD5值校验通过后, 会将该值设置为对象元数据x-obs-meta-md5checksum, 用于下载或复制时校验MD5。</li></ul> <p><b>注意</b></p> <p>如果您的业务涉及加密场景, 请勿使用该参数。</p>
p	附加参数, 可选	<p>每个分段上传任务的最大并发数, 默认为配置文件中的defaultParallels。</p>
threshold	附加参数, 可选	<p>开启分段上传任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>如果待上传的文件小于该阈值, 则使用直接上传模式; 否则使用分段上传模式。</li><li>直接上传模式不会产生断点记录文件, 不支持断点续传。</li><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
acl	附加参数, 可选	<p>同步上传文件时可指定的预定义访问策略。支持的值:</p> <ul style="list-style-type: none"><li>private</li><li>public-read</li><li>public-read-write</li><li>bucket-owner-full-control</li></ul> <p><b>说明</b></p> <p>以上四个值分别对应: 私有读写、公共读、公共读写、桶拥有者完全控制, 四种预定义访问策略。</p>
meta	附加参数, 可选	<p>上传文件时可指定的自定义元数据。格式为: key1:value1#key2:value2#key3:value3。</p> <p><b>说明</b></p> <p>以上值表示上传文件完成后, 桶内对象包含三组自定义元数据, 分别为: key1:value1、key2:value2、key3:value3。</p>
ps	附加参数, 可选	<p>每个分段上传任务的段大小, 单位: 字节, 取值范围是100KB~5GB, 默认为配置文件中的defaultPartSize。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>支持配置为auto, 此时obsutil会根据源文件大小自动设置每个分段任务的段大小。</li></ul>

参数	约束	描述
cpd	附加参数, 可选	<p>生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。</p> <p><b>说明</b> 每个分段上传任务会产生唯一对应的断点记录文件并保存至该文件夹的upload子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
j	附加参数, 同步文件夹时可选	<p>同步上传文件夹时批量任务的最大并发数, 默认为配置文件中的defaultJobs。</p> <p><b>说明</b> 工具会保证该值至少为1。</p>
exclude	附加参数, 同步文件夹时可选	<p>不包含文件的匹配模式, 如: *.txt。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持“*”匹配多个任意字符和“?”匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li> <li>您可以使用“\*”代表匹配“*”字符本身, 使用“\?”代表匹配“?”字符本身。</li> <li>如果待上传的文件名匹配该参数, 则跳过该文件的上传。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 (macOS/Linux操作系统使用单引号, Windows操作系统使用双引号) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于文件全路径 (含文件名和文件目录)。</li> <li>该匹配模式仅对文件夹中的文件生效。</li> <li>支持指定多个exclude参数, 如-exclude=*.xxx -exclude=*.xxx</li> </ul>
include	附加参数, 同步文件夹时可选	<p>包含文件的匹配模式, 如: *.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持“*”匹配多个任意字符和“?”匹配单个任意字符。</li> <li>您可以使用“\*”代表匹配“*”字符本身, 使用“\?”代表匹配“?”字符本身。</li> <li>首先执行exclude的匹配规则, 如果待上传的文件名不匹配exclude, 则判断待上传的文件名是否匹配该参数, 如果匹配则上传该文件, 否则跳过该文件的上传。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 (macOS/Linux操作系统使用单引号, Windows操作系统使用双引号) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于文件全路径 (含文件名和文件目录)。</li> <li>该匹配模式仅对文件夹中的文件生效。</li> <li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx</li> </ul>

参数	约束	描述
at	附加参数，同步上传文件夹时可选	同步上传文件时指定按照文件最后访问时间作为timeRange匹配模式，仅上传最后访问时间满足timeRange时间段的文件。 <b>说明</b> <ul style="list-style-type: none"> <li>需配合timeRange使用</li> </ul>
disableDirObject	附加参数，同步上传多文件夹时可选	同步上传文件夹时，文件夹本身不会作为一个对象上传，在有很多空文件夹场景时候可以避免无用文件夹上传到对象存储桶中；文件夹下有文件时候，文件夹下文件依然会上传并保持原有路径格式，不受影响。
timeRange	附加参数，同步上传文件夹时可选	同步上传文件时的时间段匹配模式，仅上传最后修改时间在该时间段内的文件。 该匹配模式优先级低于文件名匹配模式：exclude和include，优先执行文件名匹配模式后才会执行该匹配模式。 <b>说明</b> <ul style="list-style-type: none"> <li>使用“time1-time2”，代表匹配的时间段，其中time1必须小于等于time2，格式为yyyyMMddHHmmss。</li> <li>时间格式支持自动补全，例如yyyyMMdd等价于yyyyMMdd000000，yyyyMM等价于yyyyMM01000000。</li> <li>使用“*-time2”，代表匹配最后修改时间在time2之前的所有文件，使用“time1-*”，代表匹配最后修改时间time1之后的所有文件。</li> </ul> <b>须知</b> 该匹配模式表示的时间是UTC时间。
mf	附加参数，可选	设置名称匹配模式（include、exclude）和时间匹配模式（timeRange）对文件夹也生效。
o	附加参数，可选	生成结果清单文件的文件夹，命令执行完成后，会在该文件夹下生成结果清单文件（可能包含成功结果、失败结果和警告结果三个文件），默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"> <li>结果清单文件命名规则：sync_{succeed   failed   warning}_report_时间_TaskId.txt。</li> <li>单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024，可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li> <li>当涉及多文件/文件夹时，如需确认失败任务的具体错误信息，可参考结果清单文件夹下的失败结果清单文件"sync_failed_report_时间_TaskId.txt"及日志路径下的<b>日志文件</b>；</li> </ul>
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.12 增量同步复制对象

### 功能说明

将源桶指定路径下的所有对象同步到目标桶指定路径，使两边内容保持一致。这里的增量同步有两层含义：1) 增量，依次比较源对象和目标对象，只复制存在变化的源对象；2) 同步，命令执行完成后，保证源桶指定路径是目标桶指定路径的子集，即源桶指定路径下的所有对象均能在目标桶中找到对应对象。

#### 须知

- 同步复制对象时，请确保OBS桶中的源对象不发生变化，否则可能产生同步失败或者数据不一致的问题。
- 复制时需要有被复制对象的读权限和目标桶的写权限。
- 如果没有开启客户端跨区域复制模式，同步复制的源桶和目标桶必须在同一个区域（Region）。
- 同步复制每个源对象时会对比目标桶中对应路径的目标对象，仅在目标对象不存在，或者目标对象大小与源对象大小不一致，或者目标对象的最后修改时间早于源对象的最后修改时间时进行复制。
- 如果源桶是并行文件系统（POSIX），则目标桶不能是对象存储桶。

### 命令结构

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

#### 说明

同步复制的源路径和目标路径不能完全相同，且源路径和目标路径不能相互嵌套。

### 运行示例

- 以Windows操作系统为例，运行**obsutil sync obs://bucket-test/temp/ obs://bucket-test2/temp/**，同步复制对象。  
`obsutil sync obs://bucket-test/temp/ obs://bucket-test2/temp`  
Start at 2024-09-25 04:48:10.1147483 +0000 UTC  
Parallel: 5                   Jobs: 5  
Threshold: 50.00MB            PartSize: auto  
CheckpointDir: C:\Users\Administrator\obsutil\_checkpoint  
Task id: 104786c8-27c2-48fc-bc6a-5886596fb0e

```
OutputDir: C:\Users\Administrator\.obsutil_output  
[=====] 100.00% 10/s 0s  
Succeed count: 5      Failed count: 0  
Metrics [max cost:298 ms, min cost:192 ms, average cost:238.00 ms, average tps:9.71, transferred size:  
7.20MB]  
Task id: 0476929d-9d23-4dc5-b2f8-0a0493f027c5
```

- 更多运行示例，请参见[同步复制示例](#)。

## 参数说明

参数	约束	描述
srcbucket	必选	源桶名。
dstbucket	必选	目标桶名。
dest	可选	目标对象名前缀。 <b>说明</b> 如果该值不为空且不以“/”结尾，则工具会自动为该值末尾加上“/”作为目标对象名前缀。
key	可选	源对象名前缀。 规则如下： <ul style="list-style-type: none"><li>• 如果该值为空，则将目标桶内满足目标对象名前缀的所有对象与源桶中的所有对象保持一致；</li><li>• 如果该值不为空，则将目标桶内满足目标对象名前缀的所有对象与源桶内满足源对象名前缀的所有对象保持一致。</li></ul> <b>说明</b> <ul style="list-style-type: none"><li>• 如果该值不为空且不以“/”结尾，则工具会自动为该值末尾加上“/”作为源对象名前缀。</li><li>• 请参考<a href="#">同步复制示例</a>理解该参数的使用方式。</li></ul>
dryRun	附加参数， 可选	测试模式运行，不执行实际的复制操作。
crr	附加参数， 可选	复制时使用客户端跨区域复制模式，从通过数据流的方式从源桶直接复制数据到目标桶，且两个桶可以是任意两个OBS服务的桶。 <b>说明</b> <ul style="list-style-type: none"><li>• 如果设置了该参数，必须确保更新了配置文件中客户端跨区域复制的相关配置信息，具体可参考<a href="#">更新配置文件</a>。</li><li>• 复制时源桶对应的配置信息为配置文件中的：akCrr/skCrr/tokenCrr/endpointCrr，目标桶对应的配置信息为配置文件中的：ak/sk/token/endpoint。</li></ul> <b>须知</b> 开启跨区域复制后，因为会占用执行命令主机上传和下载的双向带宽、CPU和内存等资源，可能会影响执行命令主机的性能。

参数	约束	描述
vlength	附加参数, 可选	复制完成后, 验证目标桶对象大小是否与源桶中对象的大小一致。 <b>说明</b> 必须与crr参数配合使用。
vmd5	附加参数, 可选	复制完成后, 验证目标桶的MD5值是否与源桶中对象的MD5值一致。 <b>说明</b> <ul style="list-style-type: none"> <li>必须与crr参数配合使用。</li> <li>源桶中对象必须包含元数据x-obs-meta-md5chksum, 否则会忽略MD5值校验。MD5值校验通过后, 会将该值设置为目标对象元数据x-obs-meta-md5chksum, 用于下载或复制时校验MD5。</li> </ul> <b>注意</b> 如果您的业务涉及加密场景, 请勿使用该参数。
p	附加参数, 可选	每个分段复制任务的最大并发数, 默认为配置文件中的defaultParallels。
threshold	附加参数, 可选	开启分段复制任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"> <li>如果待复制的源对象小于该阈值, 则使用直接复制模式; 否则使用分段复制模式。</li> <li>直接复制模式不会产生断点记录文件, 不支持断点续传。</li> <li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li> </ul>
acl	附加参数, 可选	复制对象时可指定的目标对象的预定义访问策略。支持的值: <ul style="list-style-type: none"> <li>private</li> <li>public-read</li> <li>public-read-write</li> <li>bucket-owner-full-control</li> </ul> <b>说明</b> 以上四个值分别对应: 私有读写、公共读、公共读写、桶所有者完全控制, 四种预定义访问策略。
meta	附加参数, 可选	复制对象时可指定的目标对象的自定义元数据。格式为: key1:value1#key2:value2#key3:value3。 <b>说明</b> 以上值表示复制对象完成后, 桶内目标对象包含三组自定义元数据, 分别为: key1:value1、key2:value2、key3:value3。



参数	约束	描述
ps	附加参数, 可选	每个分段复制任务的段大小, 单位: 字节, 取值范围是100KB~5GB, 默认为配置文件中的defaultPartSize。 <b>说明</b> <ul style="list-style-type: none"><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>支持配置为auto, 此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。 <b>说明</b> <p>每个分段复制任务会产生唯一对应的断点记录文件并保存至该文件夹的copy子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
j	附加参数, 批量复制时 可选	同步复制时的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> <p>工具会保证该值至少为1。</p>
exclude	附加参数, 批量复制时 可选	不包含源对象的匹配模式, 如: *.txt。 <b>说明</b> <ul style="list-style-type: none"><li>支持“*”匹配多个任意字符和“?”匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li><li>您可以使用“\*”代表匹配“*”字符本身, 使用“\?”代表匹配“?”字符本身。</li><li>如果待复制的对象名匹配该参数, 则跳过该对象的复制。</li></ul> <b>须知</b> <ul style="list-style-type: none"><li>建议使用引号传递该匹配模式 (macOS/Linux操作系统使用单引号, Windows操作系统使用双引号) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li><li>该匹配模式作用于对象全路径 (含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt)。</li><li>该匹配模式仅适用于对象名非“/”结尾的对象。</li><li>支持指定多个exclude参数, 如-exclude=*.xxx -exclude=*.xxx</li></ul>

参数	约束	描述
include	附加参数, 批量复制时可选	<p>包含源对象的匹配模式, 如: *.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符。</li> <li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li> <li>首先执行exclude的匹配规则, 如果待复制的对象名不匹配exclude, 则判断待复制的对象名是否匹配该参数, 如果匹配则复制该对象, 否则跳过该文件的复制。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> <li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx</li> </ul>
timeRange	附加参数, 可选	<p>同步复制对象时的时间段匹配模式, 仅复制最后修改时间在该时间段内的对象。</p> <p>该匹配模式优先级低于对象名匹配模式: exclude和include, 优先执行对象名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>使用 “time1-time2”, 代表匹配的时间段, 其中time1必须小于等于time2, 格式为yyyyMMddHHmmss。</li> <li>时间格式支持自动补全, 例如yyyyMMdd等价于yyyyMMdd000000, yyyyMM等价于yyyyMM01000000。</li> <li>使用 “*-time2”, 代表匹配最后修改时间在time2之前的所有文件, 使用 “time1-*”, 代表匹配最后修改时间time1之后的所有文件。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>该匹配模式表示的时间是UTC时间。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> </ul>
mf	附加参数, 可选	<p>设置名称匹配模式 ( include、exclude ) 和时间匹配模式 ( timeRange ) 对对象名以 “/” 结尾的对象也生效。</p>

参数	约束	描述
o	附加参数, 可选	生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件(可能包含成功结果、失败结果和警告结果三个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"><li>结果清单文件命名规则: sync_{succeed   failed   warning}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件"sync_failed_report_时间_TaskId.txt"及日志路径下的<a href="#">日志文件</a>;</li></ul>
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.13 增量同步下载对象

### 功能说明

将OBS源桶指定路径下的所有对象同步到本地目标路径, 使两边内容保持一致。这里的增量同步有两层含义: 1) 增量, 依次比较源对象和目标文件, 只下载存在变化的源对象; 2) 同步, 命令执行完成后, 保证OBS指定源桶指定路径是本地目标路径的子集, 即OBS源桶指定路径下的所有对象均能在本地目标路径下找到对应文件。

#### 须知

- 同步下载时, 请确保OBS桶中的源对象不发生变化, 否则可能产生同步失败或者数据不一致的问题。
- 同步下载每个对象时会对比本地对应路径的文件, 仅在文件不存在, 或者文件大小与对象大小不一致, 或者文件的最后修改时间早于对象的最后修改时间时进行下载。
- 同步下载不支持单文件同步下载, 仅支持同步下载文件夹。

### 命令结构

- Windows操作系统  
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]

- macOS/Linux操作系统  
`./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]`

## 运行示例

- 以Windows操作系统为例，运行**obsutil sync obs://bucket-test/temp d:\temp**，同步下载对象。  

```
obsutil sync obs://bucket-test/temp d:\temp
Start at 2024-09-30 08:53:22.327072 +0000 UTC

Parallel: 5          Jobs: 5
Threshold: 50.00MB   PartSize: auto
VerifyLength: false  VerifyMd5: false
CheckpointDir: xxxx

Task id: 3066a4b0-4d21-4929-bb84-4829c32cbd0f
OutputDir: xxxx
TempFileDir: xxxx

[=====] 100.00% tps:17.86 155.59 KB/s
7.20MB/7.20MB 0s
Succeed count: 6      Failed count: 0
Succeed bytes: xxxx
Metrics [max cost:153 ms, min cost:129 ms, average cost:92.00 ms, average tps:17.86, transferred size:
7.20MB]

Task id: 3066a4b0-4d21-4929-bb84-4829c32cbd0f
```
- 更多运行示例，请参见[同步下载示例](#)。

## 参数说明

参数	约束	描述
folder_url	必选	本地文件夹路径。
bucket	必选	桶名。
key	可选	待同步下载的对象名前缀。 规则如下： <ul style="list-style-type: none"><li>• 如果该值为空，则将folder_url指定的文件夹下的所有内容整个桶的内容保持一致；</li><li>• 如果该值不为空，则将folder_url指定的文件夹下的内容与桶内以该值作为对象名前缀的内容保持一致。</li></ul> <b>说明</b> <ul style="list-style-type: none"><li>• 如果该值不为空且不以“/”结尾，则工具会自动为该值末尾加上“/”作为对象名前缀。</li><li>• 请参考<a href="#">同步下载示例</a>理解该参数的使用方式。</li></ul>
tempFileDir	附加参数，可选	同步下载时保存临时文件的文件夹，默认为配置文件中的defaultTempFileDir。 <b>说明</b> <ul style="list-style-type: none"><li>• 由于分段下载时生成的临时文件会保存在该目录，请确保运行obsutil的用户对该路径有写权限。</li><li>• 该路径的可用空间需要大于待下载对象的大小。</li></ul>

参数	约束	描述
dryRun	附加参数, 可选	测试模式运行, 不执行实际的下载操作。
vlength	附加参数, 可选	下载完成后, 验证本地文件大小是否与桶中对象的大小一致。
vmd5	附加参数, 可选	下载完成后, 验证本地文件的MD5值是否与桶中对象的MD5值一致。 <b>说明</b> 桶中对象必须包含元数据x-obs-meta-md5chksum, 否则会忽略MD5值校验。 <b>注意</b> 如果您的业务涉及加密场景, 请勿使用该参数。
p	附加参数, 可选	每个分段下载任务的最大并发数, 默认为配置文件中的defaultParallels。
threshold	附加参数, 可选	开启分段下载任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"><li>• 如果待下载的对象小于该阈值, 则使用直接下载模式; 否则使用分段下载模式。</li><li>• 直接下载模式不会产生断点记录文件, 不支持断点续传。</li><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
ps	附加参数, 可选	每个分段下载任务的段大小, 单位: 字节, 默认为配置文件中的defaultPartSize。 <b>说明</b> <ul style="list-style-type: none"><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>• 支持配置为auto, 此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。 <b>说明</b> 每个分段下载任务会产生唯一对应的断点记录文件并保存至该文件夹的down子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。
j	附加参数, 批量下载时可选	同步下载时的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> 工具会保证该值至少为1。

参数	约束	描述
exclude	附加参数, 批量下载时可选	<p>不包含源对象的匹配模式, 如: *.txt。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li> <li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li> <li>如果待下载的对象名匹配该参数, 则跳过该对象的复制。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> <li>支持指定多个exclude参数, 如-exclude=*.xxx -exclude=*.xxx</li> </ul>
include	附加参数, 批量下载时可选	<p>包含源对象的匹配模式, 如: *.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符。</li> <li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li> <li>首先执行exclude的匹配规则, 如果待下载的对象名不匹配exclude, 则判断待下载的对象名是否匹配该参数, 如果匹配则下载该对象, 否则跳过该文件的下载。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> <li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx</li> </ul>

参数	约束	描述
timeRange	附加参数, 可选	<p>同步下载对象时的时间段匹配模式, 仅下载最后修改时间在该时间段内的对象。</p> <p>该匹配模式优先级低于对象名匹配模式: exclude 和 include, 优先执行对象名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>使用 “time1-time2”, 代表匹配的时间段, 其中 time1 必须小于等于 time2, 格式为 yyyyMMddHHmmss。</li><li>时间格式支持自动补全, 例如 yyyyMMdd 等价于 yyyyMMdd000000, yyyyMM 等价于 yyyyMM01000000。</li><li>使用 “*-time2”, 代表匹配最后修改时间在 time2 之前的所有文件, 使用 “time1-*”, 代表匹配最后修改时间 time1 之后的所有文件。</li></ul> <p><b>须知</b></p> <ul style="list-style-type: none"><li>该匹配模式表示的时间是 UTC 时间。</li><li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li></ul>
mf	附加参数, 可选	设置名称匹配模式 (include、exclude) 和时间匹配模式 (timeRange) 对对象名以 “/” 结尾的对象也生效。
o	附加参数, 可选	<p>生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件 (可能包含成功结果、失败结果和警告结果三个文件), 默认为运行 obsutil 命令的用户目录的子文件夹 obsutil_output。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>结果清单文件命名规则: sync_{succeed   failed   warning}_report_时间_TaskId.txt。</li><li>单个结果清单文件默认情况下最大为 30MB 且最大可保留的文件个数为 1024, 可在配置文件中通过 recordMaxLogSize 和 recordBackups 分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件 “sync_failed_report_时间_TaskId.txt” 及日志路径下的 <a href="#">日志文件</a>;</li></ul>
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.14 恢复失败的上传任务

### 功能说明

根据任务号 (TaskId) 恢复失败的上传任务。

### 命令结构

- Windows操作系统

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

- macOS/Linux操作系统

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

### 运行示例

- 以Windows操作系统为例，运行**obsutil cp -recover 104786c8-27c2-48fc-bc6a-5886596fb0ed -f**，恢复失败的上传任务。

```
obsutil cp -recover 104786c8-27c2-48fc-bc6a-5886596fb0ed -f
Start at 2024-10-08 01:10:07.3809685 +0000 UTC

Parallel: 5          Jobs: 5
Threshold: 50.00MB  PartSize: auto
VerifyLength: false VerifyMd5: false
CheckpointDir: xxxx

Task id: a628d6da-c562-4a1f-b687-4fa125de0dc3
OutputDir: xxxx
TempFileDir: xxxx

[=====] 100.00% tps:35.71 2.02 KB/s
7.20MB/7.20MB 0s
Succeed count: 1    Failed count: 0
Succeed bytes: xxx
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71, transferred size:70B]

Task id: a628d6da-c562-4a1f-b687-4fa125de0dc3
```

### 参数说明

参数	约束	描述
recover	附加参数，必选	待恢复上传任务结果清单文件的任务号。 <b>说明</b> <ul style="list-style-type: none"><li>任务号 (TaskId) 可在每次上传任务运行完毕后获取，或者通过<b>结果清单</b>文件名查询 (文件名除去后缀.txt后的后36位)。</li><li>待恢复的上传任务会从结果清单的文件夹中查找，结果清单文件夹的路径参考附加参数o。</li></ul>
arcDir	附加参数，可选	上传文件成功后的归档路径，上传成功后的文件会移动到该路径下。



参数	约束	描述
dryRun	附加参数, 可选	测试模式运行, 不执行实际的上传操作。
u	附加参数, 可选	增量上传操作, 设置该参数后, 上传每个文件时会对比桶中对应路径的对象, 仅在对象不存在, 或者对象大小与文件大小不一致, 或者文件的最后修改时间晚于对象的最后修改时间时进行上传。
vlength	附加参数, 可选	上传完成后, 验证桶中对象的大小是否与本地文件大小一致。
vmd5	附加参数, 可选	上传完成后, 验证桶中对象的MD5值是否与本地文件的MD5值一致。 <b>说明</b> <ul style="list-style-type: none"><li>如果待上传的本地文件较大, 使用该参数将会因为计算MD5而导致整体性能下降。</li><li>MD5值校验通过后, 会将该值设置为对象元数据x-obs-meta-md5checksum, 用于下载或复制时校验MD5。</li></ul> <b>注意</b> <p>如果您的业务涉及加密场景, 请勿使用该参数。</p>
p	附加参数, 可选	每个分段上传任务的最大并发数, 默认为配置文件中的defaultParallels。
threshold	附加参数, 可选	开启分段上传任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"><li>如果待上传的文件小于该阈值, 则使用直接上传模式; 否则使用分段上传模式。</li><li>直接上传模式不会产生断点记录文件, 不支持断点续传。</li><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
acl	附加参数, 可选	上传文件时可指定的预定义访问策略。支持的值: <ul style="list-style-type: none"><li>private</li><li>public-read</li><li>public-read-write</li></ul> <b>说明</b> <p>以上三个值分别对应: 私有读写、公共读、公共读写, 三种预定义访问策略。</p>
meta	附加参数, 可选	上传文件时可指定的自定义元数据。格式为: key1:value1#key2:value2#key3:value3。 <b>说明</b> <p>以上值表示上传文件完成后, 桶内对象包含三组自定义元数据, 分别为: key1:value1、key2:value2、key3:value3。</p>

参数	约束	描述
ps	附加参数, 可选	<p>每个分段上传任务的段大小, 单位: 字节, 取值范围是100KB~5GB, 默认为配置文件中的defaultPartSize。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>支持配置为auto, 此时obsutil会根据源文件大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	<p>生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。</p> <p><b>说明</b></p> <p>每个分段上传任务会产生唯一对应的断点记录文件并保存至该文件夹的upload子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
f	附加参数, 可选	强制操作, 不进行询问提示。
j	附加参数, 可选	<p>上传文件夹时任务的最大并发数, 默认为配置文件中的defaultJobs。</p> <p><b>说明</b></p> <p>工具会保证该值至少为1。</p>
exclude	附加参数, 可选	<p>不包含文件的匹配模式, 如: *.txt。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>支持“*”匹配多个任意字符和“?”匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li><li>您可以使用“\*”代表匹配“*”字符本身, 使用“\?”代表匹配“?”字符本身。</li><li>如果待上传的文件名匹配该参数, 则跳过该文件的上传。</li></ul> <p><b>须知</b></p> <ul style="list-style-type: none"><li>建议使用引号传递该匹配模式 (macOS/Linux操作系统使用单引号, Windows操作系统使用双引号) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li><li>该匹配模式作用于文件全路径 (含文件名和文件目录)。</li><li>该匹配模式不适用于文件夹。</li><li>支持指定多个exclude参数, 如-exclude=*.xxx -exclude=*.xxx</li></ul>

参数	约束	描述
include	附加参数, 可选	<p>包含文件的匹配模式, 如: *.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符。</li> <li>您可以使用 “**” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li> <li>首先执行exclude的匹配规则, 如果待上传的文件名不匹配exclude, 则判断待上传的文件名是否匹配该参数, 如果匹配则上传该文件, 否则跳过该文件的上传。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 (macOS/Linux操作系统使用单引号, Windows操作系统使用双引号) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于文件全路径 (含文件名和文件目录)。</li> <li>该匹配模式不适用于文件夹。</li> <li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx</li> </ul>
at	附加参数, 可选	<p>恢复上传文件时指定按照文件最后访问时间作为timeRange匹配模式, 仅上传最后访问时间满足timeRange时间段的文件。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>需配合timeRange使用</li> </ul>
timeRange	附加参数, 可选	<p>上传文件时的时间段匹配模式, 仅上传最后修改时间在该时间段内的文件。</p> <p>该匹配模式优先级低于文件名匹配模式: exclude和include, 优先执行文件名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>使用 “time1-time2”, 代表匹配的时间段, 其中time1必须小于等于time2, 格式为yyyyMMddHHmmss。</li> <li>时间格式支持自动补全, 例如yyyyMMdd等价于yyyyMMdd000000, yyyyMM等价于yyyyMM01000000。</li> <li>使用 “*-time2”, 代表匹配最后修改时间在time2之前的所有文件, 使用 “time1-*”, 代表匹配最后修改时间time1之后的所有文件。</li> </ul> <p><b>须知</b></p> <p>该匹配模式表示的时间是UTC时间。</p>
mf	附加参数, 可选	<p>设置名称匹配模式 (include、exclude) 和时间匹配模式 (timeRange) 对对象名以 “/” 结尾的对象也生效。</p>

参数	约束	描述
o	附加参数, 可选	生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件 (可能包含成功结果、失败结果和警告结果三个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"><li>结果清单文件命名规则: cp_{succeed   failed   warning}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件"cp_failed_report_时间_TaskId.txt"及日志路径下的<b>日志文件</b>;</li></ul>
clear	附加参数, 可选	设置在恢复上传任务完成后删除已恢复任务对应的失败结果清单文件。
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.15 恢复失败的复制任务

### 功能说明

根据任务号 (TaskId) 恢复失败的复制任务。

### 命令结构

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

### 运行示例

- 以Windows操作系统为例, 运行**obsutil cp -recover=0476929d-9d23-4dc5-b2f8-0a0493f027c5 -f**, 批量复制对象。  
`obsutil cp -recover=0476929d-9d23-4dc5-b2f8-0a0493f027c5 -f`  
Start at 2024-10-08 01:10:07.3809685 +0000 UTC  
  
Parallel: 5                   Jobs: 5  
Threshold: 50.00MB           PartSize: auto  
VerifyLength: false           VerifyMd5: false

```
CheckpointDir: xxxx

Task id: a628d6da-c562-4a1f-b687-4fa125de0dc3
OutputDir: xxxx
TempFileDir: xxxx

[=====] 100.00% tps:35.71 2.02 KB/s
7.20MB/7.20MB 0s
Succeed count: 1      Failed count: 0
Succeed bytes:  xxx
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71, transferred
size:70B]

Task id: a628d6da-c562-4a1f-b687-4fa125de0dc3
```

## 参数说明

参数	约束	描述
recover	附加参数， 必选	待恢复复制任务结果清单文件的任务号。 <b>说明</b> <ul style="list-style-type: none"><li>任务号 (TaskId) 可在每次复制任务运行完毕后获取，或者通过<a href="#">结果清单</a>文件名查询 (文件名除去后缀.txt后的后36位)。</li><li>待恢复的复制任务会从结果清单的文件夹中查找，结果清单文件夹的路径参考附加参数o。</li></ul>
dryRun	附加参数， 可选	测试模式运行，不执行实际的上传操作。
crr	附加参数， 可选	复制时使用客户端跨区域复制模式，从通过数据流的方式从源桶直接复制数据到目标桶，且两个桶可以是任意两个OBS服务的桶。 <b>说明</b> <ul style="list-style-type: none"><li>如果设置了该参数，必须确保更新了配置文件中客户端跨区域复制的相关配置信息，具体可参考<a href="#">更新配置文件</a>。</li><li>复制时源桶对应的配置信息为配置文件中的：akCrr/skCrr/tokenCrr/endpointCrr，目标桶对应的配置信息为配置文件中的：ak/sk/token/endpoint。</li></ul> <b>须知</b> <p>开启跨区域复制后，因为会占用执行命令主机上传和下载的双向带宽、CPU和内存等资源，可能会影响执行命令主机的性能。</p>
vlength	附加参数， 可选	复制完成后，验证目标桶对象大小是否与源桶中对象的大小一致。 <b>说明</b> <p>必须与crr参数配合使用。</p>

参数	约束	描述
vmd5	附加参数, 可选	复制完成后, 验证目标桶的MD5值是否与源桶中对象的MD5值一致。 <b>说明</b> <ul style="list-style-type: none"><li>• 必须与crr参数配合使用。</li><li>• 源桶中对象必须包含元数据x-obs-meta-md5chksum, 否则会忽略MD5值校验。MD5值校验通过后, 会将该值设置为目标对象元数据x-obs-meta-md5chksum, 用于下载或复制时校验MD5。</li></ul> <b>注意</b> <p>如果您的业务涉及加密场景, 请勿使用该参数。</p>
u	附加参数, 可选	增量复制操作, 设置该参数后, 复制每个源对象时会对比目标桶中对应路径的目标对象, 仅在目标对象不存在, 或者目标对象大小与源对象大小不一致, 或者源对象的最后修改时间晚于目标对象的最后修改时间时进行复制。
p	附加参数, 可选	每个分段复制任务的最大并发数, 默认为配置文件中的defaultParallels。
threshold	附加参数, 可选	开启分段复制任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"><li>• 如果待复制的源对象小于该阈值, 则使用直接复制模式; 否则使用分段复制模式。</li><li>• 直接复制模式不会产生断点记录文件, 不支持断点续传。</li><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
acl	附加参数, 可选	复制对象时可指定的目标对象的预定义访问策略。支持的值: <ul style="list-style-type: none"><li>• private</li><li>• public-read</li><li>• public-read-write</li></ul> <b>说明</b> <p>以上三个值分别对应: 私有读写、公共读、公共读写, 三种预定义访问策略。</p>
meta	附加参数, 可选	复制对象时可指定的目标对象的自定义元数据。格式为: key1:value1#key2:value2#key3:value3。 <b>说明</b> <p>以上值表示复制对象完成后, 桶内目标对象包含三组自定义元数据, 分别为: key1:value1、key2:value2、key3:value3。</p>

参数	约束	描述
ps	附加参数, 可选	每个分段复制任务的段大小, 单位: 字节, 取值范围是100KB~5GB, 默认为配置文件中的defaultPartSize。 <b>说明</b> <ul style="list-style-type: none"><li>支持带容量单位配置, 例如, 配置1MB代表1048576字节;</li><li>支持配置为auto, 此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。 <b>说明</b> <p>每个分段复制任务会产生唯一对应的断点记录文件并保存至该文件夹的copy子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
f	附加参数, 可选	强制操作, 不进行询问提示。
j	附加参数, 可选	批量复制任务的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> <p>工具会保证该值至少为1。</p>
exclude	附加参数, 可选	不包含源对象的匹配模式, 如: *.txt。 <b>说明</b> <ul style="list-style-type: none"><li>支持“*”匹配多个任意字符和“?”匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li><li>您可以使用“\*”代表匹配“*”字符本身, 使用“\?”代表匹配“?”字符本身。</li><li>如果待复制的对象名匹配该参数, 则跳过该对象的复制。</li></ul> <b>须知</b> <ul style="list-style-type: none"><li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li><li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li><li>该匹配模式仅适用于对象名非“/”结尾的对象。</li><li>支持指定多个exclude参数, 如-exclude=*.xxx -exclude=*.xxx</li></ul>

参数	约束	描述
include	附加参数, 可选	<p>包含源对象的匹配模式, 如: *.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符。</li> <li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li> <li>首先执行exclude的匹配规则, 如果待复制的对象名不匹配exclude, 则判断待复制的对象名是否匹配该参数, 如果匹配则复制该对象, 否则跳过该文件的复制。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> <li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx</li> </ul>
timeRange	附加参数, 可选	<p>复制对象时的时间段匹配模式, 仅复制最后修改时间在该时间段内的对象。</p> <p>该匹配模式优先级低于对象名匹配模式: exclude和include, 优先执行对象名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>使用 “time1-time2”, 代表匹配的时间段, 其中time1必须小于等于time2, 格式为yyyyMMddHHmmss。</li> <li>时间格式支持自动补全, 例如yyyyMMdd等价于yyyyMMdd000000, yyyyMM等价于yyyyMM01000000。</li> <li>使用 “*-time2”, 代表匹配最后修改时间在time2之前的所有文件, 使用 “time1-*”, 代表匹配最后修改时间time1之后的所有文件。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>该匹配模式表示的时间是UTC时间。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> </ul>
mf	附加参数, 可选	<p>设置名称匹配模式 ( include、exclude ) 和时间匹配模式 ( timeRange ) 对对象名以 “/” 结尾的对象也生效。</p>



参数	约束	描述
o	附加参数, 可选	生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件(可能包含成功结果、失败结果和警告结果三个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"><li>结果清单文件命名规则: cp_{succeed   failed   warning}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件"cp_failed_report_时间_TaskId.txt"及日志路径下的<a href="#">日志文件</a>;</li></ul>
clear	附加参数, 可选	设置在恢复复制任务完成后删除已恢复任务对应的失败结果清单文件。
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.16 恢复失败的下载任务

### 功能说明

根据任务号 (TaskId) 恢复失败的下载任务。

### 命令结构

- Windows操作系统  
`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] [-clear] [-config=xxx]`
- macOS/Linux操作系统  
`./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] [-clear] [-config=xxx]`

### 运行示例

- 以Windows操作系统为例, 运行**obsutil cp -recover=3066a4b0-4d21-4929-bb84-4829c32cbd0f d:\ -f -r**, 批量下载对象。  
`obsutil cp -recover=3066a4b0-4d21-4929-bb84-4829c32cbd0f -f -r`  
Start at 2024-10-08 01:10:07.3809685 +0000 UTC  
  
Parallel: 5                   Jobs: 5  
Threshold: 50.00MB           PartSize: auto  
VerifyLength: false           VerifyMd5: false

```
CheckpointDir: xxxx

Task id: a628d6da-c562-4a1f-b687-4fa125de0dc3
OutputDir: xxxx
TempFileDir: xxxx

[=====] 100.00% tps:35.71 2.02 KB/s
7.20MB/7.20MB 0s
Succeed count: 1      Failed count: 0
Succeed bytes: xxx
Metrics [max cost:90 ms, min cost:45 ms, average cost:63.80 ms, average tps:35.71, transferred
size:70B]

Task id: a628d6da-c562-4a1f-b687-4fa125de0dc3
```

## 参数说明

参数	约束	描述
recover	附加参数, 必选	待恢复下载任务结果清单文件的任务号。 <b>说明</b> <ul style="list-style-type: none"> <li>任务号 (TaskId) 可在每次下载任务运行完毕后获取, 或者通过<b>结果清单</b>文件名查询 (文件名除去后缀.txt后的后36位)。</li> <li>待恢复的下载任务会从结果清单的文件夹中查找, 结果清单文件夹的路径参考附加参数o。</li> </ul>
tempFileDir	附加参数, 可选	下载时保存临时文件的文件夹, 默认为配置文件中的defaultTempFileDir。 <b>说明</b> <ul style="list-style-type: none"> <li>由于分段下载时生成的临时文件会保存在该目录, 请确保运行obsutil的用户对该路径有写权限。</li> <li>该路径的可用空间需要大于待下载对象的大小。</li> </ul>
dryRun	附加参数, 可选	测试模式运行, 不执行实际的上传操作。
u	附加参数, 可选	增量下载操作, 设置该参数后, 下载每个对象时会对比本地对应路径的文件, 仅在文件不存在, 或者文件大小与对象大小不一致, 或者对象的最后修改时间晚于文件的最后修改时间时进行下载。
vlenght	附加参数, 可选	下载完成后, 验证本地文件大小是否与桶中对象的大小一致。
vmd5	附加参数, 可选	下载完成后, 验证本地文件的MD5值是否与桶中对象的MD5值一致。 <b>说明</b> 桶中对象必须包含元数据x-obs-meta-md5chksum, 否则会忽略MD5值校验。 <b>注意</b> 如果您的业务涉及加密场景, 请勿使用该参数。
p	附加参数, 可选	每个分段下载任务的最大并发数, 默认为配置文件中的defaultParallels。

参数	约束	描述
threshold	附加参数, 可选	开启分段下载任务的阈值, 单位: 字节, 默认为配置文件中的defaultBigfileThreshold。 <b>说明</b> <ul style="list-style-type: none"><li>• 如果待下载的对象小于该阈值, 则使用直接下载模式; 否则使用分段下载模式。</li><li>• 直接下载模式不会产生断点记录文件, 不支持断点续传。</li><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li></ul>
ps	附加参数, 可选	每个分段下载任务的段大小, 单位: 字节, 默认为配置文件中的defaultPartSize。 <b>说明</b> <ul style="list-style-type: none"><li>• 支持带容量单位配置, 例如, 配置1MB代表1048576字节。</li><li>• 支持配置为auto, 此时obsutil会根据源对象大小自动设置每个分段任务的段大小。</li></ul>
cpd	附加参数, 可选	生成断点记录文件的文件夹, 默认为运行obsutil命令的用户目录的子文件夹.obsutil_checkpoint。 <b>说明</b> <p>每个分段下载任务会产生唯一对应的断点记录文件并保存至该文件夹的down子文件夹下, 分段任务执行成功后, 对应的断点记录文件会被自动删除; 分段任务执行失败或被中断后, 下次执行该分段任务时会尝试通过对应的断点记录文件恢复任务。</p>
f	附加参数, 可选	强制操作, 不进行询问提示。
j	附加参数, 可选	批量下载任务的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> <p>工具会保证该值至少为1。</p>

参数	约束	描述
exclude	附加参数, 可选	<p>不包含源对象的匹配模式, 如: *.txt。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符, 例如abc*.txt代表匹配以abc开头以.txt结尾的任意文件。</li> <li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li> <li>如果待下载的对象名匹配该参数, 则跳过该对象的复制。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> <li>支持指定多个exclude参数, 如-exclude=*.xxx -exclude=*.xxx</li> </ul>
include	附加参数, 可选	<p>包含源对象的匹配模式, 如: *.jpg。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>支持 “*” 匹配多个任意字符和 “?” 匹配单个任意字符。</li> <li>您可以使用 “\*” 代表匹配 “*” 字符本身, 使用 “\?” 代表匹配 “?” 字符本身。</li> <li>首先执行exclude的匹配规则, 如果待下载的对象名不匹配exclude, 则判断待下载的对象名是否匹配该参数, 如果匹配则下载该对象, 否则跳过该文件的下载。</li> </ul> <p><b>须知</b></p> <ul style="list-style-type: none"> <li>建议使用引号传递该匹配模式 ( macOS/Linux操作系统使用单引号, Windows操作系统使用双引号 ) 防止特殊符号被操作系统转义, 导致不可预期的结果。</li> <li>该匹配模式作用于对象全路径 ( 含从根路径开始的对象前缀和对象名, 例如, 桶内对象路径为obs://bucket/src1/src2/test.txt, 则对象的全路径为src1/src2/test.txt ) 。</li> <li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li> <li>支持指定多个include参数, 如-include=*.xxx -include=*.xxx</li> </ul>

参数	约束	描述
timeRange	附加参数, 可选	<p>下载对象时的时间段匹配模式, 仅下载最后修改时间在该时间段内的对象。</p> <p>该匹配模式优先级低于对象名匹配模式: exclude 和 include, 优先执行对象名匹配模式后才会执行该匹配模式。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>使用 “time1-time2”, 代表匹配的时间段, 其中 time1 必须小于等于 time2, 格式为 yyyyMMddHHmmss。</li><li>时间格式支持自动补全, 例如 yyyyMMdd 等价于 yyyyMMdd000000, yyyyMM 等价于 yyyyMM01000000。</li><li>使用 “*-time2”, 代表匹配最后修改时间在 time2 之前的所有文件, 使用 “time1-*”, 代表匹配最后修改时间 time1 之后的所有文件。</li></ul> <p><b>须知</b></p> <ul style="list-style-type: none"><li>该匹配模式表示的时间是 UTC 时间。</li><li>该匹配模式仅适用于对象名非 “/” 结尾的对象。</li></ul>
mf	附加参数, 可选	设置名称匹配模式 (include、exclude) 和时间匹配模式 (timeRange) 对对象名以 “/” 结尾的对象也生效。
o	附加参数, 可选	<p>生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件 (可能包含成功结果、失败结果和警告结果三个文件), 默认为运行 obsutil 命令的用户目录的子文件夹 obsutil_output。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>结果清单文件命名规则: cp_{succeed   failed   warning}_report_时间_TaskId.txt。单个结果清单文件默认情况下最大为 30MB 且最大可保留的文件个数为 1024, 可在配置文件中通过 recordMaxLogSize 和 recordBackups 分别配置。</li><li>当涉及多文件/文件夹时, 如需确认失败任务的具体错误信息, 可参考结果清单文件夹下的失败结果清单文件 “cp_failed_report_时间_TaskId.txt” 及日志路径下的 <a href="#">日志文件</a>;</li></ul>
clear	附加参数, 可选	设置在恢复下载任务完成后删除已恢复任务对应的失败结果清单文件。
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

## 5.17 列举分段上传任务

### 功能说明

查询桶内分段上传任务。

### 命令结构

- Windows操作系统  
obsutil ls obs://bucket[/prefix] [-s] [-d] -m [-a] [-uploadIdMarker=xxx] [-marker=xxx] [-limit=1] [-format=default] [-config=xxx]
- macOS/Linux操作系统  
./obsutil ls obs://bucket[/prefix] [-s] [-d] -m [-a] [-uploadIdMarker=xxx] [-marker=xxx] [-limit=1] [-format=default] [-config=xxx]

### 运行示例

- 以Windows操作系统为例，运行**obsutil ls obs://bucket-test -m -limit=10**，查询桶内分段上传任务。

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

Listing multipart uploads.

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

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

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

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

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

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

Folder number: 0
Upload number: 6
```

- 更多运行示例，请参见[列举分段上传任务示例](#)。

### 参数说明

参数	约束	描述
bucket	必选	桶名。
prefix	可选	列举分段上传任务时的对象名前缀。 <b>说明</b> 如果为空，则表示列举桶内所有分段上传任务。

参数	约束	描述
s	附加参数, 可选	以精简格式显示查询结果。 <b>说明</b> 在精简格式下, 返回结果只包含分段上传任务对象名和UploadId。
d	附加参数, 可选	仅列举当前目录下的分段上传任务和子目录, 而非递归列举所有分段上传任务和子目录。
m	附加参数, 必选	列举桶内分段上传任务。
a	附加参数, 可选	同时列举桶内对象和桶内分段上传任务。
marker	附加参数, 可选	列举桶内分段上传任务的起始位置, 返回结果是分段上传任务对象名按照字典序排序后该参数以后的所有分段上传任务。 <b>说明</b> 请参考 <a href="#">列举分段上传任务示例</a> 理解该参数的使用方式。
uploadIdMarker	附加参数, 可选	列举桶内分段上传任务的起始位置, 必须与marker配合使用。返回结果是对象名和UploadId按照字典序排序后该参数以后的所有分段上传任务。
limit	附加参数, 可选	列举结果的最大个数, 小于等于0表示列举所有结果。 <b>说明</b> 如果桶内分段上传任务较多, 建议设置该值限制每次列举的分段上传任务个数, 列举结束后, 如果仍有分段上传任务, 会在结果中返回下次请求的marker和uploadIdMarker, 用于继续列举剩余的分段上传任务。
format	附加参数, 可选	指定以自定义格式打印列举结果。当前仅支持值[default], 指定列举结果在一行显示。
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 5.18 删除分段上传任务

### 功能说明

- 通过分段上传任务的ID, 删除指定桶中的分段上传任务。
- 按指定的对象名前缀批量删除分段上传任务。

### 命令结构

- Windows操作系统
  - 删除单个分段上传任务  
`obsutil abort obs://bucket/key -u=xxx [-f] [-fr] [-o=xxx] [-config=xxx]`

- 批量删除分段上传任务  
`obsutil abort obs://bucket[key] -r [-f] [-o=xxx] [-j=1] [-config=xxx]`
- macOS/Linux操作系统
  - 删除单个分段上传任务  
`./obsutil abort obs://bucket/key -u=xxx [-f] [-fr] [-o=xxx] [-config=xxx]`
  - 批量删除分段上传任务  
`./obsutil abort obs://bucket[key] -r [-f] [-o=xxx] [-j=1] [-config=xxx]`

## 运行示例

- 以Windows操作系统为例，运行**`obsutil abort obs://bucket-test/key -u=xxx -f`**，删除单个分段上传任务。  

```
obsutil abort obs://bucket-test/key -u=xxx -f
Start at 2024-10-08 01:25:55.6771288 +0000 UTC

[-----] 100.00% tps:0.00 1/1 106ms
Succeed count: 1    Failed count: 0
Metrics [max cost:54 ms, min cost:54 ms, average cost:54.00 ms, average tps:8.77]

Task id: 4972589c-c775-41be-a288-bbee3edaaee9
```
- 以Windows操作系统为例，运行**`obsutil abort obs://bucket-test -r -f`**，批量删除桶中所有的分段上传任务。  

```
obsutil abort obs://bucket-test -r -f
Start at 2024-10-08 01:28:29.1980739 +0000 UTC

[-----] 100.00% tps:2924.55 3/3 202ms
Succeed count: 3    Failed count: 0
Metrics [max cost:148 ms, min cost:61 ms, average cost:113.33 ms, average tps:14.63]

Task id: cd2fd08e-fc31-47d9-b4b0-9f9a3376435f
```

## 参数说明

参数	约束	描述
bucket	必选	桶名。
key	删除单个分段上传任务时必选 批量删除分段上传任务时可选	待删除分段上传任务的对象名，或批量删除分段上传任务的对象名前缀。 <b>说明</b> 批量删除分段上传任务时，如果该参数为空，则表示删除桶中所有的分段上传任务。
u	附加参数，删除单个分段上传任务时必选	待删除分段上传任务的ID。 <b>说明</b> 该参数可以通过 <a href="#">列举分段上传任务</a> 获取到。
fr	附加参数，删除单个分段上传任务时可选	删除单个分段上传任务时生成结果清单文件。
f	附加参数，可选	强制操作，不进行询问提示。
r	附加参数，批量删除分段上传任务时必选	按指定的对象名前缀批量删除分段上传任务。



参数	约束	描述
j	附加参数, 批量删除分段上传任务时可选	批量删除分段上传任务的最大并发数, 默认为配置文件中的defaultJobs。 <b>说明</b> 工具会保证该值至少为1。
o	附加参数, 可选	生成结果清单文件的文件夹, 命令执行完成后, 会在该文件夹下生成结果清单文件(可能包含成功结果和失败结果两个文件), 默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。 <b>说明</b> <ul style="list-style-type: none"><li>结果清单文件命名规则: abort_{succeed   failed}_report_时间_TaskId.txt。</li><li>单个结果清单文件默认情况下最大为30MB且最大可保留的文件个数为1024, 可在配置文件中通过recordMaxLogSize和recordBackups分别配置。</li></ul>
config	附加参数, 可选	运行当前命令时的自定义配置文件, 可支持的配置参数请参考。

## 响应结果

请参考上传对象的[响应结果](#)。

# 6 辅助命令

## 6.1 更新配置文件

### 功能说明

更新配置文件 (.obsutilconfig) 中的部分配置信息，可更新的配置包括：endpoint、ak、sk、token。

关于配置文件 (.obsutilconfig) 中参数的详细说明，请参见[配置参数说明](#)。

### 命令结构

- Windows操作系统  
obsutil config -interactive [-crr] [-config=xxx]
- macOS/Linux操作系统  
./obsutil config -interactive [-crr] [-config=xxx]

### 运行示例

- 以Windows操作系统为例，运行**obsutil config -interactive**，更新默认配置文件的访问密钥和OBS服务地址。

```
obsutil config -interactive

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

Update config file successfully!
```

- 以Linux操作系统为例，运行**./obsutil config -interactive**，更新默认配置文件的访问密钥和OBS服务地址。

```
./obsutil config -interactive

Please input your ak:
xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
Please input your sk:
xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Please input your endpoint:
xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Please input your token:
xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Config file url:
/root/.obsutilconfig
Update config file successfully!
```

## 参数说明

参数	约束	描述
interactive	附加参数，可选	使用交互模式更新配置。
crr	附加参数，可选	更新配置文件中客户端跨区域复制模式的相关配置信息。 <b>说明</b> 不配置该参数，更新的配置文件中的字段为endpoint、ak、sk、token。 配置该参数后，更新的配置文件中的字段变为endpointCrr、akCrr、skCrr、tokenCrr。
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考 <a href="#">配置参数说明</a> 。 通过携带此参数并指定路径，可更新自定义配置文件的参数；不携带此参数则表示更新默认配置文件参数。

## 6.2 删除断点记录文件

### 功能说明

删除指定文件夹下的断点记录文件。

### 命令结构

- Windows操作系统  
obsutil clear [checkpoint\_dir] [-u] [-d] [-c] [-config=xxx]
- macOS/Linux操作系统  
./obsutil clear [checkpoint\_dir] [-u] [-d] [-c] [-config=xxx]

### 运行示例

- 以Windows操作系统为例，运行**obsutil clear -u**，删除默认文件夹下分段上传任务的断点记录文件。  

```
obsutil clear -u
Clear checkpoint files for uploading in folder [xxxxxx]
Start at 2024-10-08 01:49:37.6541204 +0000 UTC
[=====] 100.00% 0s
Succeed files is: 1      Failed files is: 0
```

## 参数说明

参数	约束	描述
checkpoint_dir	可选	断点记录文件的文件夹，默认为obsutil命令同级子文件夹.obsutil_checkpoint。
u	附加参数， 可选	删除所有分段上传任务的断点记录文件。 <b>说明</b> 同时会尝试删除每个断点记录文件中记录的分段上传任务。
d	附加参数， 可选	删除所有分段下载任务的断点记录文件。 <b>说明</b> 同时会尝试删除每个断点记录文件中记录的临时文件。
c	附加参数， 可选	删除所有分段复制任务的断点记录文件。 <b>说明</b> 同时会尝试删除每个断点记录文件中记录的分段复制任务。
config	附加参数， 可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考 <a href="#">配置参数说明</a> 。

### 📖 说明

u, d, c三个参数必须至少指定一个。

## 6.3 查看命令帮助

### 功能说明

查看工具支持的命令，或查看某个具体命令的帮助文档。

### 命令结构

- Windows操作系统  
obsutil help [command]
- macOS/Linux操作系统  
./obsutil help [command]

### 运行示例

- 以Windows操作系统为例，运行**obsutil help mb**，查看创建桶命令的帮助文档。  
**obsutil help mb**  
Summary:  
create a bucket with the specified parameters  
Syntax:  
obsutil mb obs://bucket [-acl=xxx] [-location=xxx] [-fs] [-config=xxx] [-i=xxx] [-k=xxx] [-t=xxx] [-e=xxx]  
Options:  
-fs  
create a bucket that supports POSIX

```
-acl=xxx  
the ACL of the bucket, possible values are [private|public-read|public-read-write]  
  
-location=xxx  
the region where the bucket is located  
  
-epid=xxx  
the enterprise project id of the bucket  
  
-kms=xxx  
the encryption id of the bucket  
  
-config=xxx  
the path to the custom config file when running this command  
  
-e=xxx  
endpoint  
  
-i=xxx  
access key ID  
  
-k=xxx  
security key ID  
  
-t=xxx  
security token
```

**摘要:**

按照用户指定的桶名和参数创建一个新桶

**语法:**

```
obsutil mb obs://bucket [-fs] [-az=xxx] [-acl=xxx] [-location=xxx] [-config=xxx]
```

**选项:**

```
-fs  
创建支持文件接口 ( POSIX ) 的桶  
  
-az=xxx  
创桶时可指定的可用区。支持的值: [multi-az]  
  
-acl=xxx  
创桶时可指定的预定义访问策略。支持的值: [private|public-read|public-read-write]  
  
-location=xxx  
桶所在的区域  
  
-config=xxx  
运行当前命令时的自定义配置文件
```

## 参数说明

参数	约束	描述
command	可选	查看某个具体命令的帮助文档，当前支持的命令如下： <ul style="list-style-type: none"><li>• abort, <a href="#">删除分段上传任务</a></li><li>• chattri, <a href="#">设置对象属性</a></li><li>• cp, <a href="#">上传对象</a>、<a href="#">复制对象</a>、<a href="#">下载对象</a></li><li>• ls, <a href="#">列举桶</a>、<a href="#">列举对象</a>、<a href="#">列举分段上传任务</a></li><li>• mb, <a href="#">创建桶</a></li><li>• mkdir, <a href="#">创建文件夹</a></li><li>• mv, <a href="#">移动对象</a></li><li>• rm, <a href="#">删除桶</a>、<a href="#">删除对象</a></li><li>• sign, <a href="#">生成对象的下载链接</a></li><li>• stat, <a href="#">查询桶属性</a>、<a href="#">查询对象属性</a></li><li>• sync, <a href="#">增量同步上传对象</a>、<a href="#">增量同步复制对象</a>、<a href="#">增量同步下载对象</a></li><li>• archive, <a href="#">归档日志文件</a></li><li>• clear, <a href="#">删除断点记录文件</a></li><li>• config, <a href="#">更新配置文件</a></li><li>• help, <a href="#">查看命令帮助</a></li><li>• version, <a href="#">查看版本号</a></li></ul>

## 6.4 查看版本号

### 功能说明

查看工具当前的版本号。

### 命令结构

- Windows操作系统  
obsutil version
- macOS/Linux操作系统  
./obsutil version

### 运行示例

以Windows操作系统为例。运行示例如下：

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

## 6.5 归档日志文件

### 功能说明

将日志文件归档到本地，或归档到指定的桶。

### 命令结构

- Windows操作系统
  - 归档到本地  
obsutil archive [file\_or\_folder\_url] [-config=xxx]
  - 归档到指定的桶  
obsutil archive obs://bucket[/key] [-config=xxx]
- macOS/Linux操作系统
  - 归档到本地  
obsutil archive [file\_or\_folder\_url] [-config=xxx]
  - 归档到指定的桶  
obsutil archive obs://bucket[/key] [-config=xxx]

### 运行示例

- 以Windows操作系统为例，运行**obsutil archive**，归档日志到工具运行的同级目录。  

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

### 参数说明

参数	约束	描述
file_or_folder_url	可选	<p>日志归档路径，规则如下：</p> <ul style="list-style-type: none"><li>• 如果该值为空，则以obsutil_log.zip为归档文件名，将日志归档到obsutil命令同级目录下；</li><li>• 如果file_or_folder_url不存在，则工具会判断其是否以“/”或“\”结尾，如果是，则会以该路径创建文件夹，并以obsutil_log.zip为归档文件名将日志归档到该新建的文件夹下；</li><li>• 如果file_or_folder_url不存在，且不以“/”或“\”结尾，则以该值作为归档文件名，将日志归档到本地；</li><li>• 如果file_or_folder_url为已存在的zip文件，则以该值作为归档文件名，将日志覆盖归档到本地；</li><li>• 如果file_or_folder_url为已存在的文件夹，则以obsutil_log.zip为归档文件名，将日志覆盖归档到file_or_folder_url指定的文件夹下。</li></ul> <p><b>说明</b> 归档文件均是以.zip结尾的压缩文件。</p>

参数	约束	描述
bucket	归档到指定的桶时必选	桶名。
key	归档到指定的桶时可选	归档日志到指定的桶时的对象名/对象名前缀。 规则如下： <ul style="list-style-type: none"><li>如果该值为空，则以obsutil_log.zip为对象名，将日志归档到桶的根目录；</li><li>如果该值不为空，如果该值以“/”结尾，则以该值作为对象名前缀归档日志，对象名为该值+obsutil_log.zip；否则，将该值作为对象名进行归档；</li></ul>
config	附加参数，可选	运行当前命令时的自定义配置文件，可支持的配置参数请参考 <a href="#">配置参数说明</a> 。

## 6.6 列举失败结果清单文件

### 功能说明

按最后修改时间列举指定文件夹中cp命令及sync命令生成的失败结果清单文件。

### 命令结构

- Windows操作系统  
`obsutil ls -failed [-limit=1000] [-o=xxx]`
- macOS/Linux操作系统  
`./obsutil ls -failed [-limit=1000] [-o=xxx]`

### 运行示例

以Windows操作系统为例。运行示例如下：

```
obsutil ls -failed
FileName                               ModTime
TaskId
cp_failed_report_xxx_ccb9f044-7768-4948-ae6d-4628ae9659a5.txt  2019-05-26 19:33:13.516813 +0800
CST   ccb9f044-7768-4948-ae6d-4628ae9659a5
sync_failed_report_xxx_59104460-d10e-42a1-9f76-03f7bd3eae4.txt  2019-05-27 14:30:19.91126 +0800
CST   59104460-d10e-42a1-9f76-03f7bd3eae4
Get 2 last modified failed report files.
```



## 参数说明

参数	约束	描述
failed	必选	按最后修改时间列举指定文件夹中cp命令及sync命令生成的失败结果清单文件。 <b>说明</b> <ul style="list-style-type: none"><li>当前不支持递归地对指定文件夹下的所有文件进行扫描，仅支持列举指定路径根目录下的失败结果清单文件。</li></ul>
limit	附加参数，可选	查询文件的最大个数，小于等于0表示列举所有结果，不设置时的默认值为1000。
o	附加参数，可选	指定结果清单文件所在的文件夹，不指定时默认为运行obsutil命令的用户目录的子文件夹.obsutil_output。

# 7 常见示例

## 7.1 上传示例

本章节案例中的所有命令均以Linux操作系统为例，提供上传操作指导。

假设本地文件夹的格式如下：

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

基于以上文件夹格式，不同上传场景的命令示例如下：

- 上传本地src1文件夹下的test3.txt文件至bucket-test桶的根目录，命令如下：  
`./obsutil cp /src1/test3.txt obs://bucket-test`  
成功上传后，桶内生成对象为：  
`./obs://bucket-test/test3.txt`
- 上传本地src1文件夹下的test3.txt文件至bucket-test桶的根目录，并且重命名为aaa.txt，命令如下：  
`./obsutil cp /src1/test3.txt obs://bucket-test/aaa.txt`  
成功上传后，桶内生成对象为：  
`./obs://bucket-test/aaa.txt`
- 上传本地src1文件夹下的test3.txt文件至bucket-test桶的src文件夹中，命令如下：  
`./obsutil cp /src1/test3.txt obs://bucket-test/src/`  
成功上传后，桶内生成对象为：  
`./obs://bucket-test/src/test3.txt`
- 递归上传本地src2文件夹中的所有文件（包括src2文件夹本身）至bucket-test桶的根目录，且上传过程中不进行询问操作，命令如下：  
`./obsutil cp /src1/src2 obs://bucket-test -r -f`  
成功上传后，桶内生成对象为：  
`obs://bucket-test/src2/`  
`obs://bucket-test/src2/test1.txt`  
`obs://bucket-test/src2/test2.txt`

- 递归上传本地src1文件夹中的所有文件和文件夹（包括src1文件夹本身）至bucket-test桶的src文件夹下，且上传过程中不进行询问操作，命令如下：

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

成功上传后，桶内生成对象为：

```
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
```

- 递归上传本地src1文件夹中的所有文件和文件夹（不包括src1文件夹本身）至bucket-test桶的src文件夹下，且上传过程中不进行询问操作，命令如下：

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

成功上传后，桶内生成对象为：

```
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
```

- 上传本地file1文件至bucket-test桶下，且上传过程中失败，断点续传，命令如下：

```
./obsutil cp /file1 obs://bucket-test/file -f
```

上传失败。结果如下：

```
./obsutil cp /file1 obs://bucket-test/file -f
```

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

```
[=====] 66.08% ?/s 3.35GB/4.88GB ?
```

再次执行同上命令断点续传。结果如下：

```
obsutil cp /file1 obs://bucket-test/file -f
```

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

```
[=====] 100% 307.42MB/s 4.88GB/4.88GB  
5.308s
```

```
Upload successfully, 4.88GB, n/a, /file1 --> obs://bucket-test/file, cost [6325], status [200], request id [xxxxx]
```

- 增量上传本地src1文件夹下的所有文件和文件夹至bucket-test桶的src目录，命令如下：

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

上传成功4个对象，1个为新增对象，执行结果出现Skip count字样。

```
./obsutil cp /src1 obs://bucket-test/src -f -r -u  
Start at 2024-10-08 02:00:18.8906532 +0000 UTC
```

```
Parallel: 5          Jobs: 5  
Threshold: 50.00MB   PartSize: auto  
VerifyLength: false  VerifyMd5: false  
CheckpointDir: xxxx  
Task id: 6a97974a-7929-4188-9736-fcd637d16584  
OutputDir: xxxx
```

```
[=====] 100% tps:0.00 ?/s 2.09KB/2.09KB  
5ms
```

```
Succeed count: 4      Failed count: 0      Skip count: 3  
Succeed bytes: 2.09KB
```

```
Metrics [max cost:6 ms, min cost:6 ms, average cost:1.50 ms, average tps:52.63, transferred size :2.09KB]
```

- 排除上传本地src1文件夹中src2及其包含的所有文件和文件夹，命令如下：

```
./obsutil cp /src1 obs://bucket-test/src -exclude "*src1/src2*" -f -r -mf
```

上传成功5个对象，上传信息中包含Exclude及内容字样。

```
./obsutil cp /src1 obs://bucket-test/src -exclude "*src1/src2*" -f -r -mf
```

```
Start at 2024-10-08 02:04:27.7752009 +0000 UTC
Parallel: 5          Jobs: 5
Threshold: 50.00MB   PartSize: auto
VerifyLength: false  VerifyMd5: false
Exclude: *src1/src2*
Include:
CheckpointDir: xxxx
OutputDir: xxxx
```

```
[=====] 100.00% tps:35.82 ?/s 5/5 2.39KB/
2.39KB 340ms
Succeed count: 3      Failed count: 0
Succeed bytes: 2.39KB
Metrics [max cost:338 ms, min cost:91 ms, average cost:240.40 ms, average tps:14.62, transferred size:2.39KB]
```

成功上传后，桶内生成对象为：

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

### 📖 说明

断点续传仅支持大文件，即：文件大小大于5GB或文件大小大于设置的threshold（默认50MB）。

## 7.2 同步上传示例

本章节案例中的所有命令均以Linux操作系统为例，提供同步上传操作指导。

假设本地文件夹的格式如下：

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

假设bucket-test桶内包含如下对象：

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

基于以上本地文件夹和桶内对象的格式，不同场景下同步上传的示例如下：

- 同步本地src1文件夹下的test3.txt文件至bucket-test桶的根目录，同步命令如下：

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

由于bucket-test桶中之前不存在test3.txt文件，因此test3.txt将直接上传至桶中，最终桶内对象为：

```
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/
```

- 同步本地src1文件夹下的所有文件和文件夹至bucket-test桶的src1文件夹中，同步命令如下：

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

最终桶内对象为：

```
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/
```

## 7.3 下载示例

本章节案例中的所有命令均以Linux操作系统为例，提供下载操作指导。

假设bucket-test桶内包含如下对象：

```
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
```

基于以上桶内对象格式，不同下载场景下的命令示例如下：

- 下载bucket-test桶中的test1.txt文件至本地的src1文件夹中，下载命令如下：

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

下载完成后，本地生成的文件为：

```
└─ src1  
  └─ test1.txt
```

- 下载bucket-test桶中的test1.txt文件至本地，如果本地不存在test.txt文件，则直接下载且下载后命名为test.txt，如果本地已存在test.txt文件，则以test1.txt内容覆盖test.txt进行下载。下载命令如下：

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

下载完成后，本地文件为：

```
└─ test.txt
```

- 递归下载bucket-test桶中的src2文件夹中的所有文件和文件夹（包括src2文件夹本身）至本地已存在的src1文件夹，且下载过程中不进行询问操作，下载命令为：

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

下载完成后，本地生成的文件为：

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

- 递归下载bucket-test桶中的src2文件夹中的所有文件和文件夹（不包括src2文件夹本身）至本地已存在的src1文件夹，且下载过程中不进行询问操作，下载命令为：

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

下载完成后，本地生成的文件为：

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

- 递归下载bucket-test桶中的所有文件至本地已存在的src0文件夹中，且下载过程中不进行询问操作，下载命令为：

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

下载完成后，本地生成的文件为：

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

- 排除下载bucket-test桶中src1下的src2及其包含的所有文件和文件夹，命令如下：

```
./obsutil cp obs://bucket-test/src1/ src1 -exclude "*"src1/src2*" -r -f -mf
```

下载成功4个对象，下载信息中包含Exclude及内容字样。

```
./obsutil cp obs://bucket-test/src1/ src1 -exclude "*"src1/src2*" -r -f -mf
```

```
Parallel: 5          Jobs: 5
Threshold: 50.00MB   PartSize: auto
VerifyLength: false  VerifyMd5: false
Exclude: *src1/src2*
Include:
CheckpointDir: xxxx
OutputDir: xxxx
TempFileDir: xxxx
```

```
[=====] 100.00% tps:87.78 ?/s 4/4 2.39KB/
2.39KB 223ms
Succeed count: 4      Failed count: 0
Succeed bytes: 2.39KB
Metrics [max cost:147 ms, min cost:77 ms, average cost:56.00 ms, average tps:8.85, transferred
size:2.39KB]
```

下载完成后，本地生成的文件为：

```
└─ src1
   └─ src3
      └─ test9.txt
      └─ test7.txt
```

## 7.4 同步下载示例

本章节案例中的所有命令均以Linux操作系统为例，提供同步下载操作指导。

假设bucket-test桶内包含如下对象：

```
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/
```

假设本地文件夹的格式如下：

```
└─ src1
   └─ test3.txt
```

基于以上桶内对象和本地文件夹的格式，不同场景下同步下载的示例如下：

- 同步bucket-test的src1文件夹中的所有文件和文件夹至本地已存在的src1文件夹中，同步命令如下：

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

同步下载成功后，本地src1文件夹中生成的内容如下：

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

## 7.5 复制示例

本章节案例中的所有命令均以Linux操作系统为例，提供复制操作指导。

假设bucket-src桶内包含如下对象：

```
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
```

基于以上对象格式，不同复制场景下的命令示例如下：

- 复制bucket-src桶中的test1.txt文件至bucket-dest桶中，复制命令如下：

```
./obsutil cp obs://bucket-src/test1.txt obs://bucket-dest
```

复制完成后，在bucket-dest桶中生成的对象如下：

```
obs://bucket-dest/test1.txt
```

- 复制bucket-src桶中的test1.txt文件内容至bucket-dest桶的test.txt文件中，复制命令如下：

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

复制完成后，在bucket-dest桶中生成的对象如下：

```
obs://bucket-dest/test.txt
```

- 复制bucket-src桶中的test1.txt文件至bucket-dest桶的test文件夹中，复制命令如下：

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

复制完成后，在bucket-dest桶中生成的对象如下：

```
obs://bucket-dest/test/test1.txt
```

- 复制bucket-src的src2文件夹下的所有文件和文件夹（包括src2文件夹本身）至bucket-dest桶中，且复制过程中不进行询问操作，复制命令如下：

```
./obsutil cp obs://bucket-src/src2 obs://bucket-dest -r -f
```

复制完成后，在bucket-dest桶中生成的对象如下：

```
obs://bucket-dest/src2/
obs://bucket-dest/src2/test8.txt
```

```
obs://bucket-dest/src2/src3/  
obs://bucket-dest/src2/src3/test9.txt
```

- 复制bucket-src的src2文件夹下的所有文件和文件夹（不包括src2文件夹本身）至bucket-dest桶中，且复制过程中不进行询问操作，复制命令如下：

```
./obsutil cp obs://bucket-src/src2 obs://bucket-dest -r -f -flat
```

复制完成后，在bucket-dest桶中生成的对象如下：

```
obs://bucket-dest/test8.txt  
obs://bucket-dest/src3/  
obs://bucket-dest/src3/test9.txt
```

## 7.6 同步复制示例

本章节案例中的所有命令均以Linux操作系统为例，提供同步复制操作指导。

假设源桶bucket-src内包含如下对象：

```
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/
```

假设目标桶bucket-dest内包含如下对象：

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

基于以上对象格式，不同场景下同步复制的命令示例如下：

- 同步bucket-src桶中src1文件夹下的文件和文件夹至bucket-dest桶的src1文件夹中，同步复制命令如下：

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

同步复制完成后，目标桶bucket-dest内的对象列表如下：

```
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/
```

## 7.7 列举示例

本章节案例中的所有命令均以Linux操作系统为例，提供列举操作指导。

假设bucket-test桶内包含如下对象：

```
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
```

基于以上对象格式，不同场景下列举对象的命令示例如下：



- 只列举bucket-test桶中的3个对象，列举命令如下：

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

返回结果按照对象名和版本号以字典序排列后，结果如下：

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

- 列举bucket-test桶中从test3.txt之后的3个对象，列举命令如下：

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

返回结果按照对象名和版本号以字典序排列后，结果如下：

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

- 只列举bucket-test桶根目录下的文件和子目录，不进行递归操作，即不列举子目录下的内容，列举命令如下：

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

返回结果按照对象名和版本号以字典序排列后，结果如下：

```
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/
```

## 7.8 列举分段上传任务示例

本章节案例中的所有命令均以Linux操作系统为例，提供列举上传任务操作指导。

假设bucket-test桶内包含如下分段上传任务：

```
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
```

- 只列举3个bucket-test桶内的分段上传任务示例，列举命令如下：

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

返回结果按照对象名字典序排列后，结果如下：

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

- 列举uploadid1之后的所有分段上传任务，列举命令如下：

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

返回结果按照对象名和UploadId字典序排列后，结果如下：

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

## 7.9 删除桶内全部分段上传任务

本章节案例中的所有命令均以Linux操作系统为例，提供删除桶内全部分段上传操作指导。

假设bucket-test桶内包含如下分段上传任务：

```
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
```

可以使用如下命令一次性删除桶内全部分段上传任务的碎片：

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

# 8 问题定位

## 8.1 问题定位简介

obsutil提供多种方式帮助用户自助进行日常问题分析和定位，具体方式及其说明如表 8-1所示。您通常需要结合多种方式以准确定位问题。

表 8-1 问题定位方式

方式	说明
日志文件	obsutil的日志文件分为工具日志和SDK日志，工具日志可以记录obsutil在运行过程中的正常/异常信息，SDK日志可以记录在请求OBS服务过程中的正常/异常信息。
结果清单	批量任务执行完成后，会生成结果清单文件，结果清单包含了成功、失败和警告三种类型的文件。
状态码	obsutil支持在进程运行结束后，根据不同的运行结果返回不同返回码，您可以根据返回码分析并处理问题。

## 8.2 日志文件

### 配置日志文件

obsutil的日志文件分为工具日志和SDK日志，可通过在.obsutilconfig配置文件中加入如下参数开启这两种日志：

- 工具日志（记录工具在运行期间产生的日志信息）：配置utilLogPath、utilLogBackups、utilLogLevel、utilMaxLogSize。
- SDK日志（记录工具调用OBS服务端接口期间产生的日志信息）：配置sdkLogPath、sdkLogBackups、sdkLogLevel、sdkMaxLogSize；

### 📖 说明

- 详细的配置参数说明请参考[配置参数说明](#)。
- utilLogPath和sdkLogPath是指日志文件的全路径，而非保存日志的文件夹。
- 如果未指定utilLogPath和sdkLogPath，则表示不开启工具日志和SDK日志，工具运行期间不会产生日志文件。
- 日志滚动后的文件名规则为：filename.log.number

### 须知

同一个用户如果开启多个obsutil进程同时运行时，可能导致日志文件并发写失败或日志内容丢失的问题，请在运行命令时加上-config参数为每个进程配置独立的配置文件。同时确保utilLogPath和sdkLogPath对每个进程均配置为不同的路径。

## 收集日志文件

您可以通过如下两种方式收集日志：

方式一：使用辅助命令[归档日志文件](#)；

方式二：从配置文件中，找到utilLogPath和sdkLogPath指定的路径，在本地文件系统中寻找对应路径下面的日志文件；

## 8.3 结果清单

### 配置结果清单

批量任务执行完成后，会生成结果清单文件，默认情况下结果清单文件保存在运行obsutil命令的用户目录的子文件夹.obsutil\_output下，用户可以通过在运行命令时指定-o附加参数来指定保存结果清单文件的文件夹。

### 查看结果清单

结果清单文件包含三种类型，成功结果清单、失败结果清单和警告结果清单，命名规则为：操作\_{succeed | failed | warning}\_report\_时间\_TaskId.txt。例如，上传文件夹后成功的结果清单文件名为：

cp\_succeed\_report\_20190417021908\_fb8c83e3-98ac-4d19-b23a-64023b1e0c34.txt，任务号（TaskId）就是 fb8c83e3-98ac-4d19-b23a-64023b1e0c34。

### 📖 说明

- 如果成功/失败/警告的个数为零，则不会产生对应的结果清单文件。
- 结果清单的任务号是每次运行生成的唯一标识号。
- 单个结果清单文件最大为30MB且最大可保留的文件个数为1024。
- 当涉及多文件/文件夹任务时，如需确认失败任务的具体错误信息，可参考结果清单文件夹下的失败结果清单文件及日志路径下的[日志文件](#)；
- 为了避免结果清单文件过多影响批量操作的效率，建议定期对.obsutil\_output文件夹下的结果清单文件归档备份至其他文件夹下；

## 8.4 返回码

通过进程等方式调用obsutil时，无法实时查看回显信息。obsutil支持在进程运行结束后，根据不同的运行结果生成不同的返回码，具体的返回码及其含义如表8-2所示。您可以通过以下方式获取最近一次运行结果的返回码，然后根据返回码分析并处理问题。

- macOS/Linux操作系统，执行以下命令获取最近一次运行结果的返回码：  
echo \$?
- Windows操作系统，执行以下命令获取最近一次运行结果的返回码：  
echo %errorlevel%

表 8-2 返回码

返回码	含义	常见场景举例
0	执行成功	上传对象成功。
1	文件不存在	执行cp上传文件，输入的文件路径不存在。
2	任务不存在	执行cp恢复失败的上传任务，指定的Task Id不存在。
3	参数错误	<ul style="list-style-type: none"><li>• 执行cp上传文件，输入不支持的附加选项；</li><li>• 执行cp下载文件，输入的cloud_url不合法。</li></ul> <p><b>说明</b> cloud_url为桶路径或对象路径，当下载桶中的所有对象时，cloud_url的格式必须为obs://bucketname；当下载指定对象时，cloud_url的格式必须为obs://bucketname/key。</p>
4	检查桶状态错误	执行cp上传文件夹，待上传的桶不存在。
5	命令初始化错误	<ul style="list-style-type: none"><li>• 加载配置文件出错；</li><li>• 执行cp上传文件夹，并指定了-o参数，但无法成功创建保存结果清单文件的文件夹。</li></ul>
6	执行错误	执行ls查询桶列表，因为网络超时而失败。
7	操作不支持	执行chattri修改对象属性，但操作不支持。
8	批量任务执行不完全成功	执行cp批量下载对象，部分对象下载失败。
9	中断错误	执行命令过程中用户手动Ctrl+C中断任务执行。
-1	其他未知错误	-

# 9 最佳实践

## 9.1 利用 obsutil 帮助命令查找工具功能

obsutil提供了帮助命令（help命令）用于查看各命令的帮助文档。以查询创建桶命令的帮助文档为例，详细的步骤如下：

**步骤1** 使用**obsutil help**命令查看当前支持的所有命令列表。

**步骤2** 根据命令列表中的文档说明，找到需要查看的命令缩写，例如创建桶命令缩写为**mb**。

**步骤3** 使用**obsutil help mb**命令，查看**mb**命令的使用方式和详细功能，如下图：

摘要：

按照用户指定的桶名和参数创建一个新桶

语法：

```
obsutil mb obs://bucket [-fs] [-az=xxx] [-acl=xxx] [-location=xxx] [-config=xxx]
```

选项：

-fs

创建支持文件接口（POSIX）的桶

-az=xxx

创桶时可指定的可用区。支持的值：[multi-az]

-acl=xxx

创桶时可指定的预定义访问策略。支持的值：[private|public-read|public-read-write]

-location=xxx

桶所在的区域

-config=xxx

运行当前命令时的自定义配置文件

**步骤4** 使用**obsutil mb obs://bucket-test -location xxx**，在xxx区域创建桶名为bucket-test的桶。

----**结束**

### 📖 说明

- 更多关于帮助命令的介绍请参考[查看命令帮助](#)。
- 您可以通过配置文件中的helpLanguage参数配置帮助命令的语言类型。例如，配置helpLanguage=Chinese代表配置帮忙命令的语言类型为中文。
- 当前支持的语种类型包括Chinese和English，默认为English。

## 9.2 利用 crontab 配置定时任务

### 操作场景

示例场景：每天21:30进入/root目录下，执行增量上传/src/src1文件夹到obs://bucket-test桶内。

### 前提条件

必须在Linux操作系统下，并且正常开启了crond（定时任务计划）服务。

### 📖 说明

运行service crond status命令可查看是否开启该服务。

### 配置步骤

**步骤1** 执行crontab -e命令，打开编辑定时任务的配置文件。

**步骤2** 进入Insert模式开始编辑配置文件。

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

### 📖 说明

假设obsutil工具位于/root目录下，上述配置的定时任务含义为：每天21:30进入/root目录下，执行增量上传/src/src1文件夹到obs://bucket-test桶内并将命令的输出内容重定向到/root下的obsutil\_crontab.log中。

**步骤3** 按Esc退出Insert模式，并输入:wq保存并退出。

**步骤4** 执行命令：`crontab -l`，查看当前用户已配置的定时任务，确认配置成功。

----结束

### 常见问题

1. 如何确定定时任务是否正在执行？
  - 执行命令：`tail /var/log/cron`，查看最近的定时任务执行记录。
  - 执行命令：`ps -ef | grep obsutil`，查看obsutil进程是否正在执行。
2. 如何强制终止正执行的定时任务？
  - a. 执行命令：`ps -ef | grep obsutil`，查看obsutil的进程信息。
  - b. 执行命令：`kill -9 PID`，强制终止该进程（PID为查询到的进程号）。

## 9.3 将 obsutil 设为系统内置命令

### 操作场景

作为外部软件，obsutil命令需要进入obsutil工具所在目录后方可运行，易用性较差。

操作系统提供内置命令，在系统启动时就将依赖的目录调入内存，这样在任意目录下都可以运行命令，提高工具易用性。

本章根据不同操作系统，分别介绍如何将obsutil设置为系统内置命令。

### 将 obsutil 设为 Windows 操作系统内置命令

**步骤1** 在cmd窗口中执行`echo %PATH%`命令查看当前系统已配置的所有PATH，选择任意一个PATH作为操作路径。

**步骤2** 执行命令：`mklink PATH/obsutil.exe OBSUTIL_PATH`，将obsutil配置为系统内置命令。

#### 📖 说明

PATH为步骤1选择的操作路径；OBSUTIL\_PATH为obsutil.exe工具的绝对路径。

**步骤3** 检验是否配置成功：在cmd窗口中执行命令`obsutil help`，出现工具帮助信息则表示配置成功。

----结束

### 将 obsutil 设为 macOS/Linux 操作系统内置命令

**步骤1** 创建obsutil工具目录，执行命令：

```
mkdir /obsutil
```

#### 📖 说明

- 如果已有此目录，可忽略该步骤。
- 必须保证执行用户是root。

**步骤2** 设置工具目录为755权限，执行命令：

```
chmod 755 /obsutil
```

#### 📖 说明

- 如果此目录权限为drwxr-xr-x，可忽略该步骤。
- 必须保证执行用户是root。

**步骤3** 拷贝obsutil工具到步骤1创建的目录下，并修改其权限为711。假设工具原路径为/home/test/obsutil，执行命令为：

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

**步骤4** 执行命令：`vi /etc/profile`，按i键进入Insert模式编辑文件，在文件末尾新增一行，内容为：`export PATH=$PATH:/obsutil`，然后按ESC键退出编辑，然后输入:`wq!`保存退出。



### 📖 说明

如果/etc/profile文件已有该新增行，可忽略此步骤。

**步骤5** 执行命令：**echo \$PATH**，查看当前环境变量，如果查询结果中包含:/obsutil则说明已有/obsutil环境变量，请直接执行下一步骤，否则执行**source /etc/profile**命令。

**步骤6** 检验是否配置成功：在任意目录下执行命令**obsutil help**，出现工具帮助信息则表示配置成功。

----结束

## 常见问题

1. 作为内置命令后如何找到obsutil配置文件路径？

obsutil命令同级目录下的.obsutilconfig文件即为工具obsutil的配置文件。您也可以运行**obsutil config**来获取配置文件路径，示例如下：

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

2. obsutil作为内置命令后如何卸载？

- Windows操作系统卸载步骤：

i. 运行**where obsutil**找到obsutil命令的路径：

```
where obsutil
E:\tools\bin\obsutil.exe
```

ii. 运行**del PATH**删除obsutil命令：

```
del E:\tools\bin\obsutil.exe
```

### 📖 说明

PATH为obsutil命令的路径，上例中为：E:\tools\bin\obsutil.exe。

- macOS/Linux操作系统卸载步骤：

i. 运行**which obsutil**找到obsutil命令的路径：

```
which obsutil
/obsutil/obsutil
```

ii. 运行**rm -rf PATH**删除obsutil命令：

```
rm -rf /obsutil/obsutil
```

### 📖 说明

PATH为obsutil命令的路径，上例中为：/obsutil/obsutil。

iii. 恢复系统环境变量：删除/etc/profile文件中设置的obsutil工具的PATH。

### 📖 说明

假设/etc/profile文件中有此行内容export PATH=\$PATH:/obsutil，则删除此行，或者有此行内容export PATH=\$PATH:/test/bin:/obsutil:/test1，则删除:/obsutil，即修改为export PATH=\$PATH:/test/bin:/test1。

3. 解决macOS/Linux操作系统运行obsutil内置命令失败的问题。

- 运行**obsutil help**命令，如果提示“Permission denied”错误时，请执行命令**chmod 755 OBSUTIL\_PATH**（OBSUTIL\_PATH为/obsutil目录下的obsutil路径），为该工具增加可执行权限；

- 如果提示“command not found”错误时，请重新登录；

- 如果提示“Cannot create parent folder for xx/.obsutilconfig, xx Permission denied”错误时，请确定该用户的家目录是否存在。

### 须知

ubuntu系统用`useradd`命令添加用户，不会默认创建用户家目录，需要手动创建，建议用`adduser`命令添加用户。

4. 解决macOS/Linux操作系统运行obsutil内置命令无法生成日志文件的问题。  
如果在配置文件中已正常配置了`sdkLogPath`和`utilLogPath`，但运行命令后无法生成日志文件，请确认运行命令的用户对`sdkLogPath`和`utilLogPath`是否有读写权限。

## 9.4 obsutil 性能调优最佳实践

obsutil对大于50MB的文件/对象默认采用分段方式进行上传、下载和复制，与分段任务相关的参数在`obsutilconfig`配置文件中，如表9-1所示。

表 9-1 多段参数

参数	描述
<code>defaultBigfileThreshold</code>	开启分段任务的阈值，如果待上传/下载/复制的文件大于该阈值，则采用分段任务的形式进行上传/下载/复制。单位：字节，默认为50MB。
<code>defaultPartSize</code>	每个分段任务的段大小，单位：字节，默认为auto。 说明 <ul style="list-style-type: none"><li>• 对于分段上传和分段复制取值范围是100KB~5GB。</li><li>• 对于分段下载取值范围无限制。</li></ul>
<code>defaultParallels</code>	每个分段任务的最大并发数，默认为5。

通常情况下使用分段任务不仅能提高传输效率，还能实现失败任务的断点续传。默认情况下，分段任务的段大小可以使用`auto`模式让obsutil自动调整。但在实际操作中，为了进一步提升上传下载性能，可根据文件大小、网络情况等因素进行调整，以获得最大传输效率并能保证传输任务顺利完成。

其中分段并发数可按照以下规则进行调整：

**`defaultParallels` = Min(CPU个数 \* 2, 目标对象大小 / `defaultPartSize`\*1.5)**

在上传、下载和复制命令中，obsutil提供`-p`和`-ps`参数用于修改分段并发数和分段大小，并最终以在命令中配置的参数值下发分段任务。如果在命令中不配置，则采用配置文件中的默认值。

其中分段并发数可按照以下规则进行调整：

**`p` = Min(CPU个数 \* 2, 目标对象大小 / `ps`\*1.5)**

同时针对批量上传/下载任务，还可以通过调整`defaultJobs` (`-j`) 最大并发任务数参数进行调优。

对于待上传/下载对象主要为小文件（通常在几M以内的文件），且文件个数较多的场景，由于文件本身较小，此时修改`defaultParallels` (`-p`) 和`defaultPartSize` (`-ps`) 这

两个参数的效果可能不明显，建议适当增大defaultJobs (-j) 参数以提升并行任务数，从而提升上传下载性能；

对于待上传/下载对象主要为大文件的场景，建议优先调整defaultParallels (-p) 和 defaultPartSize (-ps) 这两个参数的数值；同时要考虑总的并发数=并发任务数 ( defaultJobs ) \* 单任务最大并发数 ( defaultParallels ) ,总的并发数过大的情况下，可能会因为线程间资源切换及抢夺等原因，导致obsutil上传/下载/复制的性能出现下降；同时也更容易受整体网络情况波动的影响，导致部分任务失败；

#### 📖 说明

- 由于运行机资源限制，如果并发数配置过大，可能会因此线程间资源切换及抢夺等原因，导致obsutil上传/下载/复制的性能下降，所以请根据实际的文件大小和网络情况调整 defaultParallels (-p) 和 defaultPartSize (-ps) 这两个参数的数值。如果要进行压力测试，可以一开始将两个数值调低，然后慢慢调大寻找最佳值。
- 如果defaultParallels (-p) 和 defaultPartSize (-ps) 参数值过大，可能会因为网络不稳定等原因，产生EOF错误，此时请适当降低defaultParallels (-p) 和 defaultPartSize (-ps) 参数值。
- 如果进行的是批量操作，目标对象大小可以取待操作对象的平均值。
- 执行批量任务时，针对常见的网络问题如timeout、EOF等，建议通过增量操作(cp命令中的-u参数)进行重试；同时也可以根据失败的批量任务最终输出的taskID信息，通过恢复失败任务 ( cp命令中的-recover参数 ) 实现对失败任务的重试；

## 9.5 利用 obsutil 进行断点续传

obsutil在上传、下载和复制大文件时通过分段上传/下载/复制算法，可支持断点续传。您可根据具体的场景设置开启分段任务的阈值以实现上传/下载/复制任务失败或中断后的续传。您可通过两种方式指定开启分段任务的阈值：

### 方法一

指定配置文件中的配置参数：defaultBigfileThreshold，详情请参见[配置参数说明](#)。

### 方法二

在运行[上传](#)、[下载](#)、[复制](#)、[增量同步上传](#)、[增量同步下载](#)、[增量同步复制](#)时指定命令级参数：threshold。

例如：`obsutil cp d:\temp\test.txt obs://bucket-test/key -threshold=52428800`

其中

- `obsutil cp d:\temp\test.txt obs://bucket-test/key`可以将D盘temp路径下名为test.txt的文件上传至bucket-test桶中，并且重命名为key。
- 设置`-threshold=52428800`启动分段任务算法，设置分段任务的阈值为50MB。

以Windows操作系统为例，示例如下：

```
obsutil cp d:\temp\test.txt obs://bucket-test/key -threshold=52428800
```

```
Parallel: 3           Jobs: 3
Threshold: 50.00MB   PartSize: auto
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
```

### 📖 说明

- 优先级：命令级参数threshold > 配置文件参数defaultBigfileThreshold。
- 分段任务的阈值作用于单个文件/对象，如果文件/对象的大小大于该阈值才会对该文件/对象应用分段任务算法。
- 分段任务算法和断点续传是强制绑定关系，即一旦使用分段任务算法就相当于对该次任务启动了断点续传模式。

## 9.6 利用 obsutil 上传软链接

obsutil支持上传文件/文件夹时上传软链接指向的真实路径。您可以通过在运行上传/增量同步上传时指定命令级参数：link，来实现上传软链接的功能。

### 📖 说明

- obsutil支持识别文件夹软链接，如果软链接指向的真实路径是一个文件夹则会递归扫描该文件夹中的内容。
- 文件夹软链接不能形成环，否则上传会以panic的形式退出；如果不希望产生panic，请在配置文件中配置panicForSymbolicLinkCircle为false。
- 软链接和Windows操作系统上的快捷方式是两种独立的类型，obsutil不支持识别Windows操作系统的快捷方式。

## 9.7 为 obsutil 配置 HTTP 代理

您可以通过如下两种方式配置HTTP代理。

方式一，通过配置文件.obsutilconfig中的proxyUrl参数，示例：proxyUrl=http://username:password@your-proxy:8080；

方式二，通过系统环境变量HTTPS\_PROXY或HTTP\_PROXY，示例：HTTPS\_PROXY=http://username:password@your-proxy:8080。

### 📖 说明

- HTTP代理的格式一般为http://[用户名:密码@]代理服务器地址:端口号，其中用户名密码非必选。
- 如果proxyUrl参数和系统环境变量的优先级规则为：proxyUrl > HTTPS\_PROXY > HTTP\_PROXY。
- 用户名和密码里面不能出现“:”和“@”，否则解析用户名密码会出错。

## 9.8 使用 obsutil 实现客户端跨区域复制

obsutil支持使用客户端跨区域复制模式进行复制，即通过数据流的方式从源桶直接复制数据到目标桶，且两个桶可以是任意两个OBS服务的桶；借助客户端跨区域复制模式即可实现相同账户下不同区域桶或不同账户间桶内对象的复制；跨账号复制对象具体步骤如下：

**步骤1** 使用obsutil config命令配置源桶对应账号的AK、SK及endpoint信息：

- Windows操作系统  
obsutil config -interactive -crr

- Linux操作系统  

```
./obsutil config -interactive -crr
```

**步骤2** 使用obsutil config命令配置目标桶对应账号的AK、SK及endpoint信息：

- Windows操作系统  

```
obsutil config -interactive
```
- Linux操作系统  

```
./obsutil config -interactive
```

**步骤3** 调用cp命令，指定采用跨区域复制模式将源桶对象复制到目标桶：

- Windows操作系统  

```
obsutil cp obs://src-bucket obs://dst-bucket -f -r -crr
```
- Linux操作系统  

```
./obsutil cp obs://src-bucket obs://dst-bucket -f -r -crr
```

----结束

#### 说明

- 要使用跨区域复制模式，必须指定-crr参数；如果设置了该参数，必须确保更新了配置文件中客户端跨区域复制的相关配置信息，具体可参考[更新配置文件](#)。
- 复制时源桶对应的配置信息为配置文件中的：akCrr/skCrr/tokenCrr/endpointCrr，目标桶对应的配置信息为配置文件中的：ak/sk/token/endpoint；
- 对于同一账号下不同区域的桶，也可以按照上述步骤实现同一账号下跨区域复制对象的功能；同时也需要按照步骤一、二设置源桶及目标桶的账号信息。

#### 注意

使用-crr参数时，不会复制源对象的标准元数据信息，包括Cache-Control、Expires、Content-Encoding、Content-Disposition、Content-Type、Content-Language6个标准头域。

使用-crr参数进行跨区域复制时，不会复制源对象的ACL。可使用[-acl=xxx]指定目标对象的ACL，若不指定，则默认继承桶ACL权限。

## 9.9 指定 obsutil 上传下载时进行限速

obsutil支持通过在配置文件.obsutilconfig中配置rateLimitThreshold参数实现对上传/下载操作进行限速。

参数具体描述可参考[配置参数说明](#)，默认不指定即不进行限速，此时具体的上传/下载速度取决于用户侧的实际网络带宽及上传/下载任务的并发配置，具体调优方案可以参考[obsutil性能调优最佳实践](#)。

需要注意的是，rateLimitThreshold参数指定的是obsutil任务的全局限速，这意味着当使用cp、sync等命令进行批量文件的上传/下载时，最终的实际上传/下载速度上限仍然为rateLimitThreshold指定的阈值而不是批量任务的并发任务数\*rateLimitThreshold。

# 10 常见问题

## 10.1 是否支持同步删除文件

不支持。

当前obsutil工具支持同步上传本地目录到OBS桶中，同步成功后，如果用户删除了本地目录下的部分文件，再次进行增量同步上传时，工具将只会检查本地存在的文件是否需要增量进行上传，无法感知已删除文件是否需要在目标桶内进行删除。

需要注意的是，如果在上传过程中，本地目录中的文件有新增则无法保证工具最终上传的对象数目和本地目录内文件数目的一致性；如果希望同步这部分新增文件可以在上传完成后重新执行增量上传命令。

## 10.2 列举接口是否支持保存结果到本地文件

obsutil工具本身的列举相关操作不支持直接将结果保存在本地文件，但是可以通过操作系统提供的重定向操作实现将列举结果从屏显的标准输出重定向到指定的本地结果文件中；以列举桶内对象为例：

- Windows操作系统（在命令行cmd下执行该重定向操作）  
`obsutil ls obs://bucketName -format=default -limit=0 > D:/result.txt`
- Mac/Linux操作系统  
`./obsutil ls obs://bucketName -format=default -limit=0 > /root/result.txt`

### 📖 说明

请注意，在Windows系统下为了将输出结果重定向到本地文件中，需要在命令行窗口中执行对应的重定向操作而非启动obsutil可执行文件后执行重定向命令；obsutil工具本身不支持重定向操作。

## 10.3 列举桶内对象返回的大小与控制台桶内对象大小不一致

通过obsutil列举桶内全量对象时最终输出结果中还会包含对象的总的大小，如果通过列举操作返回的对象大小与通过控制台、OBSBrowser+工具查询的桶使用情况不一致时可以从如下两点进行排查：



1. OBSBrowser+和控制台均通过调用获取桶存量信息接口查询当前桶内对象个数及对象占用空间，对应obsutil中调用stat命令返回的结果；由于OBS桶存量是后台统计，因此存量会有一定的时延，不能实时更新，因此不建议对存量做实时校验。
2. 由于桶存量包含了桶内对象及桶内碎片的整体大小，而obsutil列举对象命令仅列举桶内对象不包含桶碎片，因此最终大小不包含桶内碎片的大小；可以通过OBSBrowser+或者控制台及[obsutil列举分段上传任务命令](#)查看当前桶内是否存在碎片。

## 10.4 批量任务执行失败如何确认错误信息

用户执行批量上传/下载/复制等批量操作时，最终输出结果中会显示批量任务执行成功的个数及失败的任务个数。为确认失败任务的具体错误原因，可以通过查看对应任务的失败结果清单文件及obsutil的日志文件进行确认。

批量任务执行完成后会输出taskID信息，根据taskID信息即可在obsutil\_output目录下查询到对应失败任务清单文件，失败结果清单文件命名规则如：cp\_{failed}\_report\_时间\_TaskId.txt，该文件中有各失败任务的详细错误信息。

同时还可以参考util工具的日志信息查看工具执行过程中记录的错误信息，为定位问题建议指定日志级别为DEBUG级别，具体设置方式及日志路径设置方式可参考[日志文件](#)。

## 10.5 i/o timeout、EOF 错误原因定位及解决

i/o timeout、EOF报错是两种常见的因为网络原因导致请求失败的场景，通常是由于网络波动等原因导致；常见的定位思路包括：

1. 通过ping桶域名 ( bucketName.endpoint ) 判断本地到指定访问的桶域名之间的网络是否不通；如果不通优先排查本地网络问题。
2. 如果问题出现概率较大，且域名可以ping通的情况下，可以通过修改endpoint为http协议后采用抓包软件进行网络抓包，抓包后可以具体分析实际网络链路中是否存在丢包等问题从而针对性解决。

解决思路：

1. 在网络不通的情况下优先解决本地网络问题；如需配置代理可以参考[为obsutil配置HTTP代理](#)。
2. 在偶现该问题的情况下，可以考虑重试命令以规避该问题；针对上传/下载/复制等场景，可以在执行cp时指定-u参数进行增量上传，可以避免重试批量任务中已经成功完成的部分任务。
3. 在网络情况不理想的场景下，针对上传/下载/复制等场景，可以考虑适当减小defaultParallels (-p) 和defaultJobs (-j) 参数来降低整体的并发任务数量，减少问题出现的概率。

## 10.6 批量任务进度条中出现问号的原因

问题原因：

在执行批量上传/下载任务时，如果目标目录包含大量的对象，由于要遍历所有对象才能统计该任务涉及的整体对象数目、大小等信息，因此在获取这些信息的过程中，进度条会以 ? 问号的形式进行展示。

**解决办法:**

待获取任务的整体信息后即会在进度条中展示相应任务信息。

## 10.7 是否支持多个 config 文件在同一个目录下

不支持。

当前OBSUtil工具支持配置多个不同的config文件，但是需要注意的是由于config文件的加密需要，因此不能在同一个目录下配置多个config文件，否则会造成冲突。

如果需要配置多个config文件，则需要将不同的config配置文件初始化在不同的目录下。



# A 配置参数说明

通过配置文件.obsutilconfig，您可以设置obsutil中的各项配置参数。

配置文件格式：

```
endpoint=<value>
ak=<value>
sk=<value>
token=<value>
endpointCrr=<value>
akCrr=<value>
skCrr=<value>
tokenCrr=<value>
connectTimeout=<value>
socketTimeout=<value>
maxRetryCount=<value>
maxConnections=<value>
defaultBigfileThreshold=<value>
defaultPartSize=<value>
defaultParallels=<value>
defaultJobs=<value>
defaultJobsCacheCount=<value>
rateLimitThreshold=<value>
sdkLogBackups=<value>
sdkLogLevel=<value>
sdkLogPath=<value>
sdkMaxLogSize=<value>
utilLogBackups=<value>
utilLogLevel=<value>
utilLogPath=<value>
utilMaxLogSize=<value>
writeBufferIoSize=<value>
readBufferIoSize=<value>
recordMaxLogSize=<value>
recordBackups=<value>
humanReadableFormat=<value>
showProgressBar=<value>
showStartTime=<value>
helpLanguage=<value>
defaultTempFileDir=<value>
checkSourceChange=<value>
skipCheckEmptyFolder=<value>
fsyncForDownload=<value>
memoryEconomicalScanForUpload=<value>
forceOverwriteForDownload=<value>
panicForSymbolicLinkCircle=<value>
fastFailThreshold=<value>
abortHttpStatusForResumableTasks=<value>
showBytesForCopy=<value>
```

```
proxyUrl=<value>  
faultTolerantMode=<value>
```

各参数的详细描述如表A-1所示。

表 A-1 obsutil 配置文件参数说明

参数	约束	描述	建议值
endpoint	必选	连接OBS的服务地址。可包含协议类型、域名、端口号（可选）。示例： <a href="#">https://your-endpoint:443</a> 。（出于安全性考虑，建议使用https协议，443端口号可以省略） <b>说明</b> <ul style="list-style-type: none"><li>您可以向企业管理员获取区域和终端节点信息。</li><li>如果配置的服务地址不带协议，则默认使用HTTPS协议。</li></ul>	N/A
ak	必选	访问密钥中的AK。 <b>说明</b> <ul style="list-style-type: none"><li>您可以从<a href="#">这里</a>查看如何获取访问密钥；</li><li>第一次运行obsutil后，工具会对AK进行加密以保护密钥安全。</li></ul>	N/A
sk	必选	访问密钥中的SK。 <b>说明</b> <ul style="list-style-type: none"><li>您可以从<a href="#">这里</a>查看如何获取访问密钥；</li><li>第一次运行obsutil后，工具会对SK进行加密以保护密钥安全。</li></ul>	N/A
token	可选	安全令牌，为空表示不设置安全令牌。当使用临时AK，SK访问的时候必须要带token。	N/A
endpointCrr	可选	客户端跨区域复制模式时，源桶的OBS的服务地址。可包含协议类型、域名、端口号。示例： <a href="#">https://your-endpoint:443</a> 。（出于安全性考虑，建议使用https协议，443端口号可以省略） <b>说明</b> <ul style="list-style-type: none"><li>您可以向企业管理员获取区域和终端节点信息。</li><li>如果配置的服务地址不带协议，则默认使用HTTPS协议。</li></ul>	N/A
akCrr	可选	客户端跨区域复制模式时，源桶使用的访问密钥中的AK。	N/A

参数	约束	描述	建议值
skCrr	可选	客户端跨区域复制模式时，源桶使用的访问密钥中的SK。	N/A
tokenCrr	可选	客户端跨区域复制模式时，源桶使用的安全令牌，为空表示不设置安全令牌。当使用临时访问密钥访问的时候必须要带此参数。	N/A
connectTimeout	可选	建立HTTP/HTTPS连接的超时时间，单位：秒，默认为30。	[5, 120]
socketTimeout	可选	读写数据的超时时间，单位：秒，默认为310。	[5, 600]
maxRetryCount	可选	请求的最大重试次数，默认为3。 <b>说明</b> 当请求OBS服务完成但获取到的HTTP状态码为408、5XX时，或者请求OBS服务发生超时错误时会进行重试。	[0, 5]
maxConnections	可选	配置允许打开的最大HTTP连接数。默认为1000。	N/A
defaultBigfileThreshold	可选	开启分段任务的阈值，如果待上传/下载/复制的文件大于该阈值，则采用分段任务的形式进行上传/下载/复制。单位：字节，默认为50MB。 <b>说明</b> 支持带容量单位配置，例如，配置1MB代表1048576字节。	至少大于5MB
defaultPartSize	可选	每个分段任务的段大小，单位：字节，默认为auto。 <b>说明</b> <ul style="list-style-type: none"> <li>对于分段上传和分段复制取值范围是100KB~5GB；</li> <li>对于分段下载取值范围无限制；</li> <li>支持带容量单位配置，例如，配置1MB代表1048576字节；</li> <li>如果值为auto，此时obsutil会根据源文件/对象大小自动设置每个分段任务的段大小。</li> </ul>	[9MB, 100MB]
defaultParallels	可选	每个分段任务的最大并发数，默认为5。	可参照 <a href="#">obsutil性能调优最佳实践</a> 进行配置

参数	约束	描述	建议值
defaultJobs	可选	批量任务的最大并发数，默认为5。 <b>说明</b> 批量任务包含上传文件夹、下载文件夹、复制文件夹、批量删除对象等。	[1, 50]
defaultJobsCache Count	可选	批量任务队列的缓存大小，表示最大可以缓存的任务个数，默认为1000000。 <b>说明</b> 缓存的任务个数越多，占用的内存资源越大，建议根据实际使用情况调整此参数值。	默认
rateLimitThreshold	可选	上传/下载命令的流控阈值，单位：字节/秒，默认为0表示不限流，最小值为10KB。 <b>说明</b> 支持带容量单位配置，例如，配置1MB代表1048576字节。	大于100KB
sdkLogBackups	可选	最大可保留SDK日志的文件个数，默认为10。	N/A
sdkLogLevel	可选	SDK日志的级别，支持的值： <ul style="list-style-type: none"><li>• DEBUG</li><li>• INFO</li><li>• WARN</li><li>• ERROR</li></ul> 默认为WARN。	N/A

参数	约束	描述	建议值
sdkLogPath	可选	<p>SDK日志文件的全路径，必须是一个文件名，默认为用户目录（macOS/Linux下为HOME目录；Windows下为C:\Users\&lt;用户名&gt;）的子文件夹.obsutil_log下的obssdk.log文件。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>如果该参数为空，则不会产生日志；</li><li>该路径必须是一个文件，不能为文件夹；</li><li>开启SDK日志后，请求OBS服务的正常/异常的日志记录都会保存在SDK日志文件中，可用于问题分析和定位；</li><li>请保证运行命令的用户对该路径有读写权限。</li></ul> <p><b>须知</b></p> <p>同一个用户如果开启多个obsutil进程同时运行时，可能导致日志文件并发写失败或日志内容丢失的问题，请在运行命令时加上-config参数为每个进程配置独立的配置文件。</p>	N/A
sdkMaxLogSize	可选	<p>SDK日志文件大小，单位：字节，默认为30MB。</p> <p><b>说明</b></p> <p>支持带容量单位配置，例如，配置1MB代表1048576字节。</p>	[10MB, 100MB]
utilLogBackups	可选	<p>最大可保留obsutil日志的文件个数，默认为10。</p>	N/A
utilLogLevel	可选	<p>obsutil日志的级别，支持的值：</p> <ul style="list-style-type: none"><li>DEBUG</li><li>INFO</li><li>WARN</li><li>ERROR</li></ul> <p>默认为INFO。</p>	N/A

参数	约束	描述	建议值
utilLogPath	可选	<p>obsutil日志文件的全路径，必须是一个文件名，默认为用户目录（macOS/Linux下为HOME目录；Windows下为C:\Users\&lt;用户名&gt;）的子文件夹.obsutil_log下的obsutil.log文件。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>如果该参数为空，则不会产生日志；</li><li>该路径必须是一个文件，不能为文件夹；</li><li>开启obsutil日志后，命令运行过程中产生的正常/异常的日志记录都会保存在obsutil日志文件中，可用于问题分析和定位；</li><li>请保证运行命令的用户对该路径有读写权限。</li></ul> <p><b>须知</b></p> <p>同一个用户如果开启多个obsutil进程同时运行时，可能导致日志文件并发写失败的问题，请在运行命令时加上-config参数为每个进程配置独立的配置文件。</p>	N/A
utilMaxLogSize	可选	<p>obsutil日志文件大小，单位：字节，默认为30MB。</p> <p><b>说明</b></p> <p>支持带容量单位配置，例如，配置1MB代表1048576字节。</p>	[10MB, 100MB]
writeBufferIoSize	可选	<p>下载数据的缓存大小，单位：字节，默认为65536。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>可根据实际场景设置该值，如果待下载的文件较大，可将该值设置较大；</li><li>支持带容量单位配置，例如，配置1MB代表1048576字节。</li></ul>	N/A
readBufferIoSize	可选	<p>上传数据的缓存大小，单位：字节，默认为8192。</p> <p><b>说明</b></p> <ul style="list-style-type: none"><li>可根据实际场景设置该值，上传大量小文件时，建议将该值设置较小；上传大文件时，可将该值设置较大；</li><li>支持带容量单位配置，例如，配置1MB代表1048576字节。</li></ul>	[4096, 65536]

参数	约束	描述	建议值
recordMaxLogSize	可选	<p>批量任务成功/失败/警告的结果清单文件大小，单位：字节，默认为30MB。</p> <p><b>说明</b> 支持带容量单位配置，例如，配置1MB代表1048576字节。</p>	[5MB, 100MB]
recordBackups	可选	<p>批量任务最大可保留的成功或失败的结果清单文件个数，默认为1024。</p>	N/A
humanReadableFormat	可选	<p>是否将列举对象结果和结果清单内容中的字节数转换为易读格式，为true时表示进行转换，默认为true。</p>	N/A
showProgressBar	可选	<p>是否在控制台显示进度条，为true时表示显示进度条，默认为true。</p>	N/A
showStartTime	可选	<p>是否在控制台显示开始时间，为true时表示显示，默认为true。</p>	N/A
helpLanguage	可选	<p>帮助文档的语种，当前完成国际化的语种类型：</p> <ul style="list-style-type: none"> <li>• Chinese</li> <li>• English</li> </ul> <p>默认为English。</p>	N/A
defaultTempFileDir	可选	<p>下载时保存临时文件的文件夹，默认为用户目录（macOS/Linux下为HOME目录；Windows下为C:\Users\<i>&lt;用户名&gt;</i>）的子文件夹obsutil_tempfile。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>• 由于分段下载时生成的临时文件会保存在该目录，请确保运行obsutil的用户对该路径有写权限；</li> <li>• 该路径的可用空间需要大于待下载对象的大小。</li> </ul>	N/A
checkSourceChange	可选	<p>上传/下载/复制时，是否检测源文件/对象变化，为true时表示开启。</p>	N/A
skipCheckEmptyFolder	可选	<p>下载时是否跳过检查OBS服务端的空文件夹，为true时表示跳过检查，默认为false。</p> <p><b>须知</b> 配置该参数为true，可能导致下载到本地后的目录结构与云端不一致。</p>	N/A

参数	约束	描述	建议值
fsyncForDownload	可选	<p>下载时是否强制同步内存数据到磁盘，为true时表示需要强制同步，默认为false。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>对数据可靠性要求极高的场景下可开启该参数；</li> <li>开启该参数后，会对下载性能产生较大影响，请谨慎使用。</li> </ul>	N/A
memoryEconomicScanForUpload	可选	<p>上传文件夹时，是否使用内存占用更少的扫描方式，为true时表示使用，默认为true。</p>	N/A
forceOverwriteForDownload	可选	<p>在linux/macOs操作系统上下载对象时，强制覆盖本地可执行文件（即使本地可执行文件正在运行），为true时表示进行强制覆盖，默认为true。</p>	N/A
panicForSymbolicLinkCircle	可选	<p>上传时对于检测到软链接形成环后的处理方式，为false时代表仅记录错误，为true时代表产生panic异常，默认为false。</p>	N/A
fastFailThreshold	可选	<p>批量任务遇到4XX错误后快速失败的阈值，当发生超过该阈值的4XX错误后进入快速失败流程，所有未执行或正在扫描的任务会被中断。默认为5。</p> <p><b>说明</b></p> <p>快速失败机制的目的是进行批量任务时，避免产生过多的流量，如果希望尽早启动快速失败，可将该值配置为0或-1，表示一旦遇到4XX错误则立即进入快速失败流程。</p>	N/A
abortHttpStatusForResumableTasks	可选	<p>分段上传/下载/复制任务快速中断的HTTP状态码，当某个分段子任务的HTTP状态码包含在该值范围内时，则会立即中断整个分段上传/下载/复制任务。默认值为：401,403,404,405,409。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>可包含多个HTTP状态码，并以逗号分隔，例如：401,403,404；</li> <li>必须是4XX的HTTP状态码，其他状态码将被忽略。</li> </ul>	默认
showBytesForCopy	可选	<p>桶间复制对象时进度条是否以字节方式显示速率，默认为false。</p>	N/A



参数	约束	描述	建议值
proxyUrl	可选	HTTP代理配置, 示例: http://username:password@your-proxy:8080。 <b>说明</b> <ul style="list-style-type: none"><li>• HTTP代理的格式一般为 http://[用户名:密码@]代理服务器地址:端口号, 其中用户名密码非必选。</li><li>• 如果proxyUrl参数和系统环境变量的优先级规则为: proxyUrl &gt; HTTPS_PROXY &gt; HTTP_PROXY。</li><li>• 用户名和密码里面不能出现“:”和“@”, 否则解析用户名密码会出错。</li></ul>	N/A
faultTolerantMode	可选	是否忽略读操作引发的panic, 默认为false, 表示不忽略。	N/A

#### 说明

- 建议值为N/A的表示需要根据实际情况进行设置;
- 建议指定sdkLogPath和utilLogPath开启SDK日志和obsutil日志;
- defaultBigfileThreshold、defaultPartSize、rateLimitThreshold、sdkMaxLogSize、utilMaxLogSize、recordMaxLogSize、readBufferIoSize、writeBufferIoSize支持带容量单位配置, 例如, 配置1MB代表1048576字节。

# B 修订记录

---

发布日期	修订记录
2024-04-15	第一次正式发布。