ALIBABA CLOUD

Alibaba Cloud

Object Storage Service

Tools

Document Version: 20220106

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.OSS tools 08
2.ossutil 11
2.1. Overview 11
2.2. Download and installation 13
2.3. Common commands 15
2.3.1. appendfromfile 15
2.3.2. bucket-cname 17
2.3.3. bucket-encryption 19
2.3.4. bucket-policy 21
2.3.5. bucket-tagging 27
2.3.6. bucket-versioning 29
2.3.7. cat 31
2.3.8. config 33
2.3.9. cors 36
2.3.10. cors-options 38
2.3.11. cp 40
2.3.11.1. Overview 40
2.3.11.2. Upload objects 40
2.3.11.3. Download objects 49
2.3.11.4. Copy objects 55
2.3.12. create-symlink (create a symbolic link) 62
2.3.13. du (query object sizes) 64
2.3.14. getallpartsize 66
2.3.15. hash 67
2.3.16. help 68
2.3.17. inventory (configure bucket inventories) 69

2.3.18. lifecycle 72
2.3.19. listpart 76
2.3.20. logging 78
2.3.21. lrb 81
2.3.22. ls 82
2.3.23. mb (create buckets) 89
2.3.24. mkdir 92
2.3.25. object-tagging (add, modify, query, and delete object... 93
2.3.26. probe 97
2.3.27. read-symlink (query a symbolic link) 106
2.3.28. referer 108
2.3.29. replication 111
2.3.30. request-payment (configure pay-by-requester) 116
2.3.31. restore (restore objects) 118
2.3.32. revert-versioning 121
2.3.33. rm 123
2.3.34. set-acl (configure or modify object or bucket ACLs) 127
2.3.35. set-meta 130
2.3.36. sign (generate signed object URLs) 134
2.3.37. stat 136
2.3.38. sync (synchronize objects) 139
2.3.38.1. Overview 139
2.3.38.2. Synchronize local files to OSS 139
2.3.38.3. Synchronize OSS objects to your computer 145
2.3.38.4. Synchronize objects between OSS paths 150
2.3.39. update 156
2.3.40. website 156
2.3.41. worm (manage retention policies) 161

2.4. View options 164
2.5. FAQ 174
3.ossbrowser 177
3.1. Install and log on to ossbrowser 177
3.2. Use ossbrowser 180
3.3. Permission management 181
3.4. Troubleshooting 184
4.ossimport 186
4.1. Architectures and configurations 186
4.2. Standalone deployment 199
4.3. Distributed deployment 203
4.4. Troubleshooting 207
5.ossfs 216
5.1. Overview 216
5.2. Installation 217
5.3. Advanced configurations 219
5.4. Common options 223
5.5. FAQ 226
6.ossftp 233
6.1. Overview 233
6.2. Installation 233
6.3. Use cases 238
6.3.1. How to store remote attachments from Discuz! to OSS 239
6.3.2. How to store remote attachments from PHPWind to O... 240
6.3.3. How to store remote attachments from WordPress web... 242
6.4. FAQ 243
7.osscmd (unavailable) 245
7.1. Overview 245

7.2. Quick start 245
7.3. Commands for operations on buckets 250
7.4. Commands for operations on objects 253
7.5. Commands for operations on parts 259
8.RAM Policy Editor 262

1.OSS tools

The following table describes the tools that can be used to streamline OSS operations.

Tools	Overview
ossbrowser	A graphical object management tool. • Provides an easy-to-use graphical interface. • Provides features similar to those of Windows Explorer. • Allows you to browse objects. • Allows you to upload and download directories (folders). • Allows you to use concurrent upload and resumable upload to upload objects. • Allows you to configure authorization policies and grant permissions to RAM users. • Supports Windows, Linux, and macOS. Limits: • The transmission speed and performance of ossbrower are not as good as those of ossutil because ossbrower is a graphical tool. • Only objects smaller than 5 GB can be moved or replicated. • Objects larger than 48.8 TB cannot be uploaded.
ossutil	A command-line tool used to manage objects and buckets. • Provides a wide range of simple and convenient commands to manage objects and buckets while high performance of operations is ensured. • Allows you to use concurrent upload and resumable upload to upload objects. • Allows you to upload and download directories (folders).
osscmd (unavailable)	A command-line tool used to manage objects and buckets. • Provides a wide range of commands to manage objects and buckets. • Supports Windows and Linux. Limits: • osscmd is compatible only with Python 2.5, 2.6, and 2.7. osscmd is incompatible with Python 3.x. • osscmd does not allow you to configure new features such as the storage class of Infrequent Access (IA) or Archive, Cold Archive, cross-region replication (CRR), and mirroring-based back-to-origin. **Notice** The osscmd operation commands have been integrated into ossutil. osscmd has been unavailable since July 31, 2019. Alibaba Cloud regrets the inconvenience caused, and appreciates your patience and understanding.

Tools	Overview
ossfs	A tool used to attach a bucket to the local file system. After you attach OSS buckets to the local file system of Linux, you can perform operations on the objects in OSS to access or share these objects by using the local file system. • Supports most features of the POSIX-compliant file systems, including file reading/writing, directories, link-related operations, permissions, UID or GID, and extended attributes. • Allows you to use multipart upload to upload large objects. • Supports MD5 verification to ensure data integrity. Limits: • Archive and Cold Archive buckets cannot be attached to local file systems by using ossfs. • If you use ossfs to edit an object that was uploaded, the object is uploaded again. • The performance of metadata-related operations such as `list directory` is compromised because you must remotely access the OSS server. • Errors may occur if you rename an object or a folder. Operation failures may cause data inconsistencies. • ossfs is unsuitable for scenarios that require highly concurrent read and write operations. • If an OSS bucket is attached to multiple clients, you are responsible for maintaining data consistency. We recommend that you schedule the time when your users can use objects, which prevents multiple clients from writing to the same object at the same time. • Hard links are not supported.
ossftp	An FTP-based tool used to manage objects in OSS. • You can use FTP clients such as FileZilla, WinSCP, and FlashFXP to manage objects in OSS. • ossftp is an FTP server that receives FTP requests and performs operations on objects and folders in OSS. • ossftp is based on Python 2.7 and later. • ossftp supports Windows, Linux, and macOS.
ossimport	A tool used to synchronize data to OSS. • Allows you to synchronize data from a third-party data source to OSS. • Supports distributed deployment. You can use multiple servers to migrate data simultaneously. • Supports migration of more than terabytes of data. • Supports Windows and Linux. • Applicable to Java 7 and Java 8.

Tools	Overview
RAM Policy Editor	A tool used to automatically generate OSS-related authorization policies. We recommend that you use this tool to generate custom authorization policies. • Generates an authorization policy based on the specified requirements. You can call this operation to create a custom policy. Then, the policy can be added to the custom policy in the RAM console. • Supports the following browsers: Google Chrome, Firefox, and Safari.

2.ossutil

2.1. Overview

ossutil allows you to manage Object Storage Service (OSS) data by using command lines on Windows, Linux, and macOS operating systems.

Install ossutil

For more information about how to download and install ossutil, see Download and installation.

Common commands

The following table describes common commands that are supported by ossutil.

Command	Description
appendfromfile	Appends content to existing append objects.
bucket-encryption	Adds, modifies, queries, or deletes encryption configurations for a bucket.
bucket-policy	Adds, modifies, queries, or deletes bucket policy configurations for a bucket.
bucket-tagging	Adds, modifies, queries, or deletes tagging configurations for a bucket.
bucket-versioning	Adds or queries versioning configurations for a bucket.
cat	Exports object content to ossutil.
config	Creates a configuration file to store information that is required to access OSS.
cors	Adds, modifies, queries, or deletes cross-origin resource sharing (CORS) configurations for a bucket.
cors-options	Tests whether a bucket allows a specified cross-origin request.
cp	Uploads, downloads, or copies objects.
create-symlink	Creates a symbolic link.
du	Queries the storage usage of a specified bucket, object, or directory.
getallpartsize	Queries the size of each part of incomplete multipart upload tasks in a bucket and the total size of these parts.
hash	Calculates the CRC-64 or MD5 hash of a local file.
help	Queries help information about a command. We recommend that you run the **help** command to query information about how to run a specified command.
inventory	Adds, queries, lists, or deletes inventory configurations for a bucket.

Command	Description
lifecycle	Adds, modifies, queries, or deletes lifecycle configurations for a bucket.
listpart	Lists the parts generated in an incomplete multipart upload task initiated for an object.
logging	Adds, modifies, queries, or deletes logging configurations for a bucket.
lrb	Lists the buckets that are located within a region or multiple regions.
ls	Lists buckets, objects, or parts.
mb	Creates a bucket.
mkdir	Creates a directory in a bucket.
object-tagging	Adds, modifies, queries, or deletes tagging configurations for an object.
probe	Monitors access to OSS. You can also run this command to troubleshoot issues that are caused by network faults or incorrect parameter settings during the upload and download process.
read-symlink	Reads the description of a symbolic link.
referer	Adds, modifies, queries, or deletes hotlink protection configurations for a bucket.
replication	Manages the Cross-Region Replication (CRR) configurations of a bucket.
request-payment	Configures the pay-by-requester mode for or queries the pay-by-requester configurations of a bucket.
restore	Restores an object from the frozen state to the readable state.
revert-versioning	Recovers a deleted object to the most recent version of the object.
rm	Deletes buckets, objects, or parts.
set-acl	Configures the access control list (ACL) for a bucket or an object.
set-meta	Configures the metadata for an uploaded object.
sign	Generates a signed URL for an object and shares the signed URL with third parties for downloads or previews.
stat	Obtains the description of a specified bucket or object.
sync	Synchronizes local files to OSS, OSS objects to local disks, or objects between OSS paths.
update	Updates the ossutil version.

Command	Description
website	Adds, modifies, queries, or deletes static website hosting, redirection, or back-to-origin configurations for a bucket.
worm	Queries the retention policies configured for a bucket.

2.2. Download and installation

ossutil supports the following operating systems: Windows, Linux, and macOS. You can download and install the ossutil version that best suits your requirements.

Version and runtime environment

- Current version: 1.7.7
- Source code: ossutil
- Runtime environment
  - Windows/Linux/macOS
  - Supported architectures: x86 (32-bit and 64-bit) and ARM (32-bit and 64-bit)

Download URLs

- Linux x86 32bit
- Linux x86 64bit
- Windows x86 32bit
- Windows x86 64bit
- macOS x86 32bit
- macOS x86 64bit
- ARM 32bit
- ARM 64bit

Download the package based on your operating system and run the corresponding binary file to install and configure ossutil. In this topic, ossutil is installed on 64-bit operating systems.

Install ossutil on Linux

1. Run the following command to download the ossutil installation package:

   ```
   wget http://gosspublic.alicdn.com/ossutil/1.7.7/ossutil64
   ```

   Notice When you use the copied URL in the wget command to download ossutil, delete the `?spm=xxxx` part from the URL.

2. Run the following command to modify the execution permissions of the file:

   ```
   chmod 755 ossutil64
   ```

3. Generate a configuration file in interactive mode.

i. Run the following command:

```
./ossutil64 config
```

ii. Configure the path of the configuration file as prompted.

We recommend that you use the default path for the configuration file by pressing the Enter key.

```
Enter the name of the configuration file. The file name can contain a path. The def
ault path is /home/user/.ossutilconfig. If you press the Enter key without specifyi
ng a different path, the file is generated in the default path.
If you want to generate the file in another path, set the --config-file option to t
he path when you run this command.
```

By default, ossutil uses */home/user/.ossutilconfig* as the configuration file. If you set the path of the configuration file, you must add the -c option to specify the configuration file each time you run this command. For example, if you save the configuration file as */home/config*, add the -c option in the following format when you run the ls command:

```
./ossutil64 ls oss://examplebucket -c /home/config
```

iii. Set the language of ossutil as prompted.

```
Enter the language: CH or EN. The default language is CH. The configuration of this
parameter takes effect after the config command is run.
```

iv. Configure the parameters, including endpoint, accessKeyID, accessKeySecret, and stsToken as prompted.

You can refer to the following descriptions when you configure the parameters:

- endpoint: Enter the endpoint of the region in which your bucket is located. For more information about the endpoint of each region, see Regions and endpoints.

  You can also add `http://` or `https://` to specify the protocol that ossutil uses to access Object Storage Service (OSS). The default protocol is HTTP. For example, if you want to access a bucket in the China (Shenzhen) region by using HTTPS, set the endpoint to `https://oss-cn-shenzhen.aliyuncs.com`.

- accessKeyID and accessKeySecret: Enter the AccessKey pair of your Alibaba Cloud account.
  - For more information about how to obtain the AccessKey pair of an Alibaba Cloud account or a Resource Access Management (RAM) user, see Obtain an AccessKey pair.
  - For more information about how to obtain the AccessKey pair provided by Security Token Service (STS) for a temporarily authorized account, see Use a temporary access credential provided by STS to access OSS.
- stsToken: This option is required only when you use a temporary STS token to access an OSS bucket. Otherwise, you can leave this parameter empty. For more information about how to generate an STS token, see Authorized third-party upload.

> **Note** For more information about configuration files, see config.

Install ossutil on Windows

1. Click the download URL listed in the preceding section to download the installation package of ossutil.
2. Decompress the downloaded installation package. Then, run the *ossutil.bat* file.
3. Run the following command to generate a configuration file:

```
D:\ossutil>ossutil64.exe config
```

4. Configure the configuration file as prompted. You can configure the configuration file in the same way as you generate configuration files in Linux. For more information, see Generate a configuration file in interactive mode.

Install ossutil on macOS

1. Run the following command to download the ossutil installation package:

```
curl -o ossutilmac64 http://gosspublic.alicdn.com/ossutil/1.7.7/ossutilmac64
```

2. Run the following command to modify the execution permissions of the file:

```
chmod 755 ossutilmac64
```

3. Run the following command to generate configurations:

```
./ossutilmac64 config
```

4. Configure the configuration file as prompted. You can configure the configuration file in the same way as you generate configuration files in Linux. For more information, see Generate a configuration file in interactive mode.

Install ossutil on ARM systems

1. Run the following command to download the ossutil installation package:

```
wget http://gosspublic.alicdn.com/ossutil/1.7.7/ossutilarm64
```

2. Run the following command to modify the execution permissions of the file:

```
chmod 755 ossutilarm64
```

3. Run the following command to generate a configuration file:

```
./ossutilarm64 config
```

4. Configure the configuration file as prompted. You can configure the configuration file in the same way as you generate configuration files in Linux. For more information, see Generate a configuration file in interactive mode.

2.3. Common commands

2.3.1. appendfromfile

This topic describes how to run the **appendfromfile** command to append content to an object that is uploaded by using append upload.

> **Notice**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about append upload, see Append upload.

Command syntax

```
./ossutil64 appendfromfile localfilename oss://bucketname/objectname [--meta ]
```

The following table describes the parameters that you can configure when you run the appendfromfile command.

Parameter	Description
localfilename	The full path of the local file that you want to upload.
bucketname	The name of the bucket that stores the object to which you want to append content.
objectname	The name of the object to which you want to append content. When you run the appendfromfile command for append upload, you can retain the original name of the local file for the object or specify another name.
--meta	The metadata of the object. You can configure this parameter only the first time you run the appendfromfile command to upload an object. Example: `--meta "x-oss-object-acl:private"`. For objects whose metadata is configured, you can run the set-meta command to modify their metadata.

Examples

In the following examples, the appendfromfile command is first used to upload a local file named exampleobject.txt in the root directory to a bucket named examplebucket. Then, the appendfromfile command is used to append content to exampleobject.txt multiple times.

1. Run the following command to upload exampleobject.txt, and then set the access control list (ACL) of the uploaded object to private:

   ```
   ./ossutil64 appendfromfile exampleobject.txt oss://examplebucket/exampleobject.txt --me
   ta "x-oss-object-acl:private"
   ```

   If a similar output is displayed, exampleobject.txt is uploaded to the specified bucket and the object size is 5 bytes.

   ```
   total append 5(100.00%) byte,speed is 0.00(KB/s)
   local file size is 5,the object new size is 5,average speed is 0.04(KB/s)
   ```

2. Run the following command to append the content of a file named dest.txt to exampleobject.txt.

If you want to append more content to exampleobject.txt, run the following command and replace dest.txt with the name of the file whose content you want to append to exampleobject.txt:

```
./ossutil64 appendfromfile dest.txt oss://examplebucket/exampleobject.txt
```

If a similar output is displayed, the content is appended to exampleobject.txt. The object size becomes 150 bytes after the append upload.

```
total append 150(100.00%) byte,speed is 0.00(KB/s)
local file size is 150,the object new size is 150,average speed is 1.19(KB/s)
```

Common options

If you use ossutil to manage buckets that are located in different regions, you can include the -e option to use the endpoint of the region in which the specified bucket is located. If you use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, include the -i option to use the AccessKey ID of the specified Alibaba Cloud account, and then include the -k option to use the AccessKey secret of the specified Alibaba Cloud account.

For example, you can run the following command to upload a file named exampleobject.txt to a bucket named examplebucket, which is located in the China (Shanghai) region and is owned by another Alibaba Cloud account:

```
./ossutil64 appendfromfile exampleobject.txt oss://examplebucket/exampleobject.txt -e shang
hai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other options that you can use for the appendfromfile command, see Common options.

2.3.2. bucket-cname

To query the custom domain names mapped to a bucket and the status of the domain names, run the **bucket-cname** command to view the CNAME configurations of the bucket.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 bucket-cname --method get oss://bucketname [local_xml_file]
```

The following table describes the parameters that you can configure when you run the bucket-cname command.

Parameter	Description
bucketname	The name of the bucket whose CNAME configurations you want to query.

Parameter	Description
local_xml_file	The name of the local file used to store the CNAME configurations. Example: `localfile.txt`. If this parameter is not specified, the CNAME configurations that are obtained are directly displayed.

Examples

- You can run the following command to query the CNAME configurations of the examplebucket bucket and write the configurations into the `localfile.txt` local file:

  ```
  ./ossutil64 bucket-cname --method get oss://examplebucket localfile.txt
  ```

  If a similar output is displayed, the CNAME configurations of the bucket are obtained:

  ```
  0.212407(s) elapsed
  ```

- You can run the following command to query the CNAME configurations of the examplebucket bucket and specify that the output is displayed without being stored in the local file:

  ```
  ./ossutil64 bucket-cname --method get oss://examplebucket
  ```

  If a similar output is displayed, the CNAME configurations of the bucket are obtained. The following result shows that the example.com and example.org domain names are mapped to the examplebucket bucket and the domain names are enabled:

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <ListCnameResult>
    <Bucket>examplebucket</Bucket>
    <Owner>148562088256****</Owner>
    <Cname>
      <Domain>example.com</Domain>
      <LastModified>2021-08-26T07:25:12.000Z</LastModified>
      <Status>Enabled</Status>
      <IsPurgeCdnCache>true</IsPurgeCdnCache>
    </Cname>
    <Cname>
      <Domain>example.org</Domain>
      <LastModified>2021-08-26T07:25:26.000Z</LastModified>
      <Status>Enabled</Status>
      <IsPurgeCdnCache>true</IsPurgeCdnCache>
    </Cname>
  </ListCnameResult>
  0.164039(s) elapsed
  ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to query the CNAME configurations of the examplebucket bucket that is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 bucket-cname --method get oss://testbucket  -e oss-cn-hangzhou.aliyuncs.com -i
LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the bucket-cname command, see Common options.

2.3.3. bucket-encryption

After you configure server-side encryption, Object Storage Service (OSS) encrypts uploaded objects and permanently stores the encrypted objects. When you download objects, OSS decrypts the objects and returns the decrypted objects. This topic describes how to run the **bucket-encryption** command to add, modify, query, or delete encryption configurations for a bucket.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about how server-side encryption works, see Server-side encryption.

Add or modify bucket encryption configurations

- Command syntax

  ```
  ./ossutil64 bucket-encryption --method put oss://bucketName  --sse-algorithm algorithmNam
  e  [--kms-masterkey-id  keyid]
  ```

  The following table describes the parameters that you can configure when you run this command to add or modify bucket encryption configurations.

Parameter	Description
bucketName	The bucket for which you want to configure server-side encryption.
--sse-algorithm	The encryption method for the bucket. Valid values: ○ KMS: The keys managed by Key Management Service (KMS) are used for encryption and decryption (SSE-KMS). ○ AES256: The keys managed by OSS are used for encryption and decryption (SSE-OSS).
--kms-masterkey-id	When the encryption method is set to SSE-KMS, OSS uses the default KMS-managed customer master key (CMK) to encrypt objects. To use the specified KMS-managed CMK to encrypt objects, set this parameter to the valid CMK ID.

- Examples
  - You can run the following command to set the default encryption method to SSE-OSS and the encryption algorithm to AES-256 for examplebucket:

    ```
    ./ossutil64 bucket-encryption --method put oss://examplebucket --sse-algorithm AES256
    ```

  - You can run the following command to set the default encryption method to SSE-KMS for examplebucket. Specify a CMK ID, and set the encryption algorithm to AES-256:

    ```
    ./ossutil64 bucket-encryption --method put oss://examplebucket --sse-algorithm KMS --km
    s-masterkey-id 9468da86-3509-4f8d-a61e-6eab1eac****
    ```

  - If a similar output is displayed, server-side encryption is configured for examplebucket:

    ```
    0.856895(s) elapsed
    ```

Query the server-side encryption configurations of a bucket

- Command syntax

  ```
  ./ossutil64 bucket-encryption --method get oss://bucket
  ```

- Examples

  You can run the following command to query the encryption configurations of examplebucket:

  ```
  ./ossutil64 bucket-encryption --method get oss://examplebucket
  ```

  If a similar output is displayed, the server-side encryption method configured for examplebucket is SSE-KMS, the CMK ID is not specified, and the encryption algorithm is AES-256:

  ```
  SSEAlgorithm:KMS
  KMSMasterKeyID:
  KMSDataEncryption:
  ```

Delete the server-side encryption configurations of a bucket

- Command syntax

  ```
  ./ossutil64 bucket-encryption --method delete oss://bucket
  ```

- Examples

  You can run the following command to delete the server-side encryption configurations of examplebucket:

  ```
  ./ossutil64 bucket-encryption --method delete oss://examplebucket
  ```

  If a similar output is displayed, server-side encryption configurations are deleted for examplebucket:

  ```
  0.856686(s) elapsed
  ```

Common options

To use command-line tool ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use command-line tool ossutil to manage buckets that are owned by multiple Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to set the encryption method to AES-256 for a bucket named examplebucket, which is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 bucket-encryption --method put oss://examplebucket --sse-algorithm AES256 -e os
s-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the bucket-encryption command, see Common options.

2.3.4. bucket-policy

Bucket policies are resource-based authorization policies. Bucket owners can use bucket policies to authorize other users to access the specified resource in Object Storage Service (OSS). This topic describes how to run the **bucket-policy** command to add, modify, query, or delete bucket policy configurations for a bucket.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about bucket policies, see Configure bucket policies to authorize other users to access OSS resources.

Add or modify bucket policies

Before you add or modify bucket policies for a bucket, you must create a JSON file on your local device, and configure bucket policies in the JSON file. You can configure multiple bucket policies in a single JSON file. However, the total size of the bucket policies cannot exceed 16 KB.

When you add or modify bucket policies, ossutil reads bucket policies from the JSON file and adds the policies to the specified bucket. When you add bucket policies, existing bucket policies are overwritten.

- Command syntax

  ```
  ./ossutil64 bucket-policy --method put oss://bucketname local_json_file
  ```

  The following table describes the parameters that you can configure when you run this command to add or modify bucket policies.

Parameter	Description
bucketname	The name of the bucket for which you want to add or modify bucket policies.
local_json_file	The name of the local JSON file in which you configure bucket policies.

- Examples
  i. Create a file named local_json_file on your local device and write different bucket policies based on different scenarios.

     The following examples show how to configure common bucket policies:

     - Specify that only anonymous requests from the specified IP address are allowed to access all resources in a bucket named examplebucket.

```
{
    "Statement": [
        {
            "Action": [
                "oss:GetObject",
                "oss:GetObjectAcl",
                "oss:ListObjects",
                "oss:RestoreObject",
                "oss:GetVodPlaylist",
                "oss:ListObjectVersions",
                "oss:GetObjectVersion",
                "oss:GetObjectVersionAcl",
                "oss:RestoreObjectVersion"
            ],
            "Condition": {
                "IpAddress": {
                    "acs:SourceIp": [
                        "10.10.10.10"
                    ]
                }
            },
            "Effect": "Allow",
            "Principal": [
                "*"
            ],
            "Resource": [
                "acs:oss:*:1746495857602745:examplebucket/*"
            ]
        },
        {
            "Action": [
                "oss:ListObjects",
                "oss:GetObject"
            ],
            "Condition": {
                "StringLike": {
                    "oss:Prefix": [
                        "*"
                    ]
                },
                "IpAddress": {
                    "acs:SourceIp": [
                        "10.10.10.10"
                    ]
                }
            },
```

```
            "Effect": "Allow",
            "Principal": [
                "*"
            ],
            "Resource": [
                "acs:oss:*:1746495857602745:examplebucket"
            ]
        }
    ],
    "Version": "1"
}
```

- Grant the specified Resource Access Management (RAM) user read-only permissions on the `hanghzou/2020` and `hanghzou/2015` directories in a bucket named examplebucket.

```
{
    "Statement": [
        {
            "Action": [
                "oss:GetObject",
                "oss:GetObjectAcl",
                "oss:ListObjects",
                "oss:RestoreObject",
                "oss:GetVodPlaylist",
                "oss:ListObjectVersions",
                "oss:GetObjectVersion",
                "oss:GetObjectVersionAcl",
                "oss:RestoreObjectVersion"
            ],
            "Effect": "Allow",
            "Principal": [
                "202147604049359142"
            ],
            "Resource": [
                "acs:oss:*:174649585760****:examplebucket/hanghzou/2020/*",
                "acs:oss:*:174649585760****:examplebucket/hangzhou/2015/*"
            ]
        },
        {
            "Action": [
                "oss:ListObjects",
                "oss:GetObject"
            ],
            "Condition": {
                "StringLike": {
                    "oss:Prefix": [
                        "hanghzou/2020/*",
                        "hangzhou/2015/*"
                    ]
                }
            },
            "Effect": "Allow",
            "Principal": [
                "202147604049359142"
            ],
            "Resource": [
                "acs:oss:*:174649585760****:examplebucket"
            ]
        }
    ],
    "Version": "1"
}
```

- Reject anonymous requests to all the objects in the `hangzhou/2021/` directory of a bucket named examplebucket.

```
{
    "Statement": [
        {
            "Action": [
                "oss:RestoreObject",
                "oss:ListObjects",
                "oss:AbortMultipartUpload",
                "oss:PutObjectAcl",
                "oss:GetObjectAcl",
                "oss:ListParts",
                "oss:DeleteObject",
                "oss:PutObject",
                "oss:GetObject",
                "oss:GetVodPlaylist",
                "oss:PostVodPlaylist",
                "oss:PublishRtmpStream",
                "oss:ListObjectVersions",
                "oss:GetObjectVersion",
                "oss:GetObjectVersionAcl",
                "oss:RestoreObjectVersion"
            ],
            "Effect": "Deny",
            "Principal": [
                "*"
            ],
            "Resource": [
                "acs:oss:*:174649585760****:examplebucket/hangzhou/2021/*"
            ]
        },
        {
            "Action": [
                "oss:ListObjects",
                "oss:GetObject"
            ],
            "Condition": {
                "StringLike": {
                    "oss:Prefix": [
                        "hangzhou/2021/*"
                    ]
                }
            },
            "Effect": "Deny",
            "Principal": [
                "*"
            ],
            "Resource": [
                "acs:oss:*:174649585760****:examplebucket"
            ]
        }
    ],
    "Version": "1"
}
```

ii. Add a bucket policy to examplebucket.

```
./ossutil64 bucket-policy --method put oss://examplebucket local_json_file
```

If a similar output is displayed, the bucket policy is added to examplebucket:

```
1.125101(s) elapsed
```

Query bucket policies

- Command syntax

```
./ossutil64 bucket-policy --method get oss://bucketname local_json_file
```

Parameter	Description
bucketname	The name of the bucket whose policies you want to query.
local_json_file	The local JSON file that is used to store the obtained bucket policies. If this parameter is not specified, obtained bucket policies are displayed without being stored in the JSON file.

- Examples

  You can run the following commands to query the bucket policies configured for a bucket named examplebucket:

```
./ossutil64 bucket-policy --method get oss://examplebucket
```

  If a similar output is displayed, the bucket policies of examplebucket are obtained and written to the local JSON file:

```
0.212407(s) elapsed
```

Delete bucket policies

If you no longer need to use bucket policies to authorize other users to access your OSS resources, delete the configured bucket policies.

- Command syntax

```
./ossuitl64 bucket-policy --method delete oss://bucketname
```

- Examples

  You can run the following command to delete all bucket policies configured for a bucket named examplebucket:

```
./ossutil64 bucket-policy --method delete oss://examplebucket
```

  If a similar output is displayed, all bucket policies configured for examplebucket are deleted:

```
0.530750(s) elapsed
```

Common options

When you use ossutil to manage buckets that are located in different regions, you can add the -e option to use the endpoint of the region in which the specified bucket is located. When you use ossutil to manage buckets that are owned by multiple Alibaba Cloud accounts, you can add the -i option to commands to use the AccessKey ID of the specified Alibaba Cloud account and add the -k option to use the AccessKey secret of the specified Alibaba Cloud account.

For example, you can run the following command to configure a bucket policy for a bucket named examplebucket, which is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 bucket-policy --method put oss://examplebucket local_json_file -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the bucket-policy command, see Common options.

2.3.5. bucket-tagging

Object Storage Service (OSS) allows you to configure bucket tagging to classify and manage buckets. For example, you can use this feature to list buckets that have specific tags and configure access control lists (ACLs) for buckets that have specific tags. This topic describes how to run the **bucket-tagging** command to add, modify, query, or delete tagging configurations for a bucket.

> **Notice**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about bucket tagging, see Bucket tagging.

Add tags to a bucket or modify the tags of a bucket

Bucket tagging uses a key-value pair as a tag to identify buckets. Each bucket can have up to 10 tags. Only the bucket owner and users who are granted the PutBucketTags permission can add tags to the bucket or modify the tags of the bucket. If other users add tags to the bucket or modify tags of the bucket, 403 Forbidden is returned with error code AccessDenied.

- Command syntax

  ```
  ./ossutil64 bucket-tagging --method put oss://bucketname key#value
  ```

  The following table describes the parameters that you can configure when you run this command to add tags to a bucket or modify the tags of a bucket.

Parameter	Description
bucketname	The name of the bucket to which you want to add a tag or of which you want to modify a tag.

Parameter	Description
key#value	The key-value pair that contains the tag information. ○ The key and value of the tag are separated by a number sign (#). The key and value of the tag must be encoded in UTF-8. ○ Each tag must have a key. The key of a tag can be up to 64 characters in length and cannot start with `http://` , `https://` , or `Aliyun` . ○ The maximum length of the value of a tag is 128 characters. The value can be empty.

If a bucket has no tags, you can run this command to add tags to the bucket. If a bucket has tags, you can run this command to overwrite existing tags.

- Examples

  You can run the following command to add two tags to a bucket named examplebucket. One tag has a key of tag1 and a value of test1, and the other tag has a key of tag2 and a value of test2.

```
./ossutil64 bucket-tagging --method put oss://examplebucket  tag1#test1 tag2#test2
```

If a similar output is displayed, the tags are added to the bucket:

```
0.300600(s) elapsed
```

Query bucket tags

- Command syntax

```
./ossutil64 bucket-tagging --method get oss://bucketname
```

- Examples

  You can run the following command to query the tags of a bucket named examplebucket:

```
./ossutil64 bucket-tagging --method get oss://examplebucket
```

If a similar output is displayed, examplebucket is configured with two tags. One tag has the key of tag1 and value of test1, and the other tag has the key of tag2 and value of test2.

```
index     tag key       tag value
-------------------------------------------------
0         "tag1"        "test1"
1         "tag2"        "test2"
0.283359(s) elapsed
```

Remove bucket tags

- Command syntax

```
./ossutil64 bucket-tagging --method delete oss://bucketname
```

- Examples

  You can run the following command to remove all tags of a bucket named examplebucket:

```
./ossutil64 bucket-tagging --method delete oss://examplebucket
```

If a similar output is displayed, all tags of examplebucket are removed:

```
0.530750(s) elapsed
```

Common options

To use command-line tool ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use command-line tool ossutil to manage buckets that are owned by multiple Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to configure tags for a bucket named examplebucket, which is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 bucket-tagging--method put oss://examplebucket key#value -e oss-cn-hangzhou.ali
yuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the bucket-tagging command, see Common options.

2.3.6. bucket-versioning

Object Storage Service (OSS) allows you to configure versioning for a bucket to protect objects stored in the bucket. After you enable versioning for a bucket, data that is overwritten or deleted in the bucket is saved as a previous version. Versioning allows you to recover a previous version of an object to protect the object from being accidentally overwritten or deleted. This topic describes how to run the **bucket-versioning** command to configure or query the versioning status of a bucket.

> **Notice**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about versioning, see Overview.

Configure the versioning status of a bucket

- Command syntax

```
./ossutil64 bucket-versioning --method put oss://bucketname versioning
```

The following table describes the parameters that you can configure when you run this command to configure the versioning status of a bucket.

Parameter	Description
bucketname	The name of the bucket whose versioning status you want to configure.

Parameter	Description
versioning	The versioning status of the bucket that you want to configure. Valid values: ◦ *enabled*: OSS enables versioning for the bucket. When an object is uploaded to a bucket that has versioning enabled, OSS generates a random string as the globally unique version ID of the object. For more information about how to manage objects in a versioned bucket, see Manage objects in a versioning-enabled bucket. ◦ *suspended*: OSS suspends versioning for the bucket. When an object is uploaded to a bucket for which versioning is suspended, OSS generates a string null as the version ID of the object. For more information about how to manage objects in a bucket for which versioning is suspended, see Manage objects in a versioning-suspended bucket. **Notice** By default, the versioning status of a bucket is disabled. After versioning is enabled for a bucket, the versioning status of the bucket cannot be set back to disabled. However, you can suspend versioning for a versioned bucket.

- Examples

  You can run the following command to enable versioning for a bucket named examplebucket:

  ```
  ./ossutil64 bucket-versioning --method put oss://examplebucket enabled
  ```

  You can run the following command to suspend versioning for a bucket named examplebucket:

  ```
  ./ossutil64 bucket-versioning --method put oss://examplebucket suspended
  ```

  If a similar output is displayed, the versioning status of the bucket named examplebucket is configured:

  ```
  0.261209(s) elapsed
  ```

Query the versioning status of a bucket

- Command syntax

  ```
  ./ossutil64 bucket-versioning --method get oss://bucketname
  ```

- Examples

  You can run the following command to query the versioning status of a bucket named examplebucket:

  ```
  ./ossutil64 bucket-versioning --method get oss://examplebucket
  ```

  If a similar output is displayed, versioning is enabled for the bucket:

  ```
  bucket versioning status:Enabled
  0.218001(s) elapsed
  ```

  If a similar output is displayed, versioning is suspended for the bucket named examplebucket:

```
bucket versioning status:Suspended
0.168791(s) elapsed
```

If a similar output is displayed, versioning is disabled for the bucket named examplebucket:

```
bucket versioning status:Null
0.158691(s) elapsed
```

Related operations

- OSS manages an object that is uploaded to a versioned bucket and an unversioned bucket in the same way. However, OSS generates a globally unique version ID for an object that is uploaded to a versioned bucket. For more information, see Upload objects.
- After you enable versioning for a bucket, objects that are overwritten or deleted in the bucket are saved as previous versions. You can specify a version ID to download the specified version of an object. For more information, see Download objects. You can specify a version ID to recover the specified previous version of an object. For more information, see Copy objects.

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by multiple Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to enable versioning for a bucket named examplebucket, which is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 bucket-versioning--method put oss://examplebucket enabled -e oss-cn-hangzhou.al
iyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the bucket-versioning command, see Common options.

2.3.7. cat

This topic describes how to run the **cat** command to view the content of a specified object in a bucket.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 cat oss://bucketname/objectname [--payer <value>] [--version-id <value>]
```

> **Notice** We recommend that you run this command to view only the content of a TXT object.

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	Specifies the name of the bucket.
objectname	Specifies the name of the object.
--payer	Specifies the payer of the traffic and request fees charged to view the content of the object. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees charged during queries, set this parameter to *requester*.
--version-id	Specifies the ID of the specified version of the object. This parameter applies only to objects in buckets for which versioning is enabled or suspended.

Examples

- View the content of the test.txt object in the examplebucket bucket for which versioning is not enabled.

  ```
  ./ossutil64 cat oss://examplebucket/test.txt
  ```

  The following output indicates the content of the test.txt object and the time it took to return the output:

  ```
  My Website Home Page.
  0.088092(s) elapsed
  ```

- View the content of the exampleobject.txt object in the examplebucket bucket for which versioning is enabled.

  ```
  ./ossutil64 cat oss://examplebucket/exampleobject.txt --version-id  CAEQARiBgID8rumR2hYiI
  GUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3****
  ```

  For more information about how to query all versions of an object, see ls.

  The following output indicates the content of a specified version of the test.txt object and the time it took to return the output:

  ```
  Hello World.
  0.044820(s) elapsed
  ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to switch to the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, if you want to view the content of the exampleobject1.txt object in the examplebucket1 bucket that belongs to another Alibaba Cloud account in the China (Shanghai) region, you can run the following command:

```
./ossutil64 cat oss://examplebucket1/exampleobject1.txt -e oss-cn-shanghai.aliyuncs.com -i
LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options of cat, see Common options.

2.3.8. config

This topic describes how to run the **config** command to create a configuration file that is used to store Object Storage Service (OSS) access information. You can add the *-c* option when you run other commands. This way, ossutil uses configurations in the specified configuration file to access OSS.

Notice Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

You can run the command in interactive or non-interactive mode. In interactive mode, you must complete related configurations by using a configuration file before you run the config command. When you run the config command, ossutil directly reads configurations from the configuration file. In non-interactive mode, you must specify configuration items when you run the config command. Compared with the non-interactive mode, the interactive mode provides higher security.

Run the command in non-interactive mode by using the following command syntax:

```
./ossutil64 config
[-e, --endpoint <value>]
[-i, --access-key-id <value>]
[-k, --access-key-secret <value>]
[-t, --sts-token <value>]
[--ram-role-arn <value>]
[-L language <value>]
[--output-dir <value>]
[-c, --config-file <value>]
```

The following table describes the configuration items.

Configuration item	Description
-e, --endpoint	The endpoint of the region in which the bucket is located. For more information, see Regions and endpoints. To specify the protocol that ossutil uses to access OSS, add `http://` or `https://`. The default protocol is HTTP.
-i, --access-key-id	The AccessKey ID in the [Credentials] section of the configuration file. For more information about how to view the AccessKey ID information, see Obtain an AccessKey pair.
-k, --access-key-secret	The AccessKey secret in the [Credentials] section of the configuration file. For more information about how to view the AccessKey secret information, see Obtain an AccessKey pair.

Configuration item	Description
-t, --sts-token	The Security Token Service (STS) token used to access OSS. You need to configure this item only when you use STS for temporary access to OSS. For more information about how to generate an STS token, see the "Temporary access credential" section in Authorized third-party upload.
--ram-role-arn	The Alibaba Cloud Resource Name (ARN) of the Resource Access Management (RAM) role used to authenticate requests. You need to configure this item only when you call AssumeRole to access OSS as a RAM user.
-L language	The language that is used by ossutil. Default value: CH. Valid values: • *CH*: Chinese. If you plan to set this configuration item to CH, make sure that your system supports UTF-8 encoding. • *EN*: English.
--output-dir	The directory in which the output objects are located. Output objects include report objects generated due to errors that occur when you run the **cp** command to copy multiple objects. Default value: the *ossutil_output* directory in the current directory.
-c, --config-file	The configuration file path of ossutil. ossutil reads the configuration file during startup.

Examples

- Generate a configuration file in interactive mode

```
./ossutil64 config
Enter the name of the configuration file. The file name can contain a path. The default p
ath is /home/user/.ossutilconfig. If you press Enter without specifying a different desti
nation, the file will be generated in the default path.
If you want to generate the file in another path, set the --config-file option to the pat
h.
If you do not enter the path of the configuration file, the default configuration file /h
ome/user/.ossutilconfig is used.
If you press the Enter key without configuring the following parameters, these parameters
are ignored. To obtain more information about the parameters, run the help config command
.
Enter the endpoint: https://oss-cn-shenzhen.aliyuncs.com
Enter the AccessKey ID: yourAccessKeyID
Enter the AccessKey secret: yourAccessKeySecret
Enter the STS token: yourStsToken
```

- Generate a configuration file in non-interactive mode

```
./ossutil64 config -e oss-cn-beijing.aliyuncs.com -i LTAIbZcdVCmQ**** -k D26oqKBudxDRBg8W
uh2EWDBrM0****  -L CH -c /myconfig
```

If you specify options other than *-L language* and *-c, --config-file* when you run the config command, the non-interactive mode is used. Then, you must use options to specify all configuration items.

Modify the configuration file

You can modify OSS access information by modifying the generated configuration file. The configuration file of ossutil is in the following format:

```
[Credentials]
        language = CH
        endpoint = oss.aliyuncs.com
        accessKeyID = your_accesskey_id
        accessKeySecret = your_accesskey_secret
        stsToken = your_sts_token
        outputDir = your_output_dir
        ramRoleArn = your_ram_role_arn
[Bucket-Endpoint]
        bucket1 = endpoint1
        bucket2 = endpoint2
        ...
[Bucket-Cname]
        bucket1 = cname1
        bucket2 = cname2
        ...
[AkService]
        ecsAk=http://10.255.255.254/latest/meta-data/Ram/security-credentials/EcsRamRoleTes
ting
```

- Bucket-Endpoint: Specify an endpoint for each specified bucket.
- Bucket-Cname: Configure a CNAME for each specified bucket. For more information about CNAMEs, see Map custom domain names.
- AkService: This item is required if you need to use a RAM role bound to an Elastic Compute Service (ECS) instance to perform operations on OSS. When you configure this item, you need only to set EcsRamRoleTesting to the name of the RAM role bound to the ECS instance. After you configure this item, you can ignore the AccessKey ID, AccessKey Secret, and STSToken items. If you configure AccessKey ID, the AkService configuration does not take effect. The AccessKey ID, AccessKey Secret, and STSToken configurations are used to verify your identity. For more information about how to bind a RAM role to an ECS instance, see Attach an instance RAM role.

> **Notice**
>
> - In the later version of ossutil, you do not need to specify Bucket-Endpoint or Bucket-Cname if you use the interactive mode. You can specify an endpoint or a CNAME for each bucket in the configuration file.
> - ossutil allows you to specify endpoints by using different methods. The following endpoint configurations take effect in descending order: endpoints specified by the --endpoint option in the command, endpoints specified in [Bucket-Cname], endpoints specified in [Bucket-Endpoint], and endpoints specified in [Credentials].

2.3.9. cors

Cross-origin resource sharing (CORS) is a standard cross-origin solution provided by HTML5 to allow web application servers to control cross-origin access. This way, the security of data transmission across origins is ensured. This topic describes how to run the **cors** command to add, modify, query, or delete CORS configurations for a bucket.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about CORS, see Configure CORS in Object Storage Service (OSS) Developer Guide.

Add or modify CORS configurations for a bucket

If no CORS configurations are added for a bucket, you can run the cors command to add CORS configurations for a bucket. If CORS configurations are added for a bucket, you can run the cors command to overwrite the existing CORS configurations.

To add or modify CORS configurations for a bucket, perform the following steps:

1. Create a local file. Specify the CORS rules that you want to add in the local file.
2. Use ossutil to read CORS configurations from the local file, and then add the CORS configurations for the specified bucket.

To add or modify CORS configurations for a bucket, use the following command syntax:

- Command syntax

```
./ossutil64 cors --method put oss://bucketname local_xml_file
```

The following table describes the parameters that you can configure when you run the cors command.

Parameter	Description
bucketname	The name of the bucket for which you want to add or modify CORS configurations.
local_xml_file	The name of the local file where CORS rules are configured. Example: `localfile.txt`.

- Examples

  i. Create a file named `localfile.txt` on the local computer and configure different CORS rules in the file.

  In the following example, AllowedOrigin is set to `www.aliyun.com`, AllowedMethod is set to `PUT`, and MaxAgeSeconds is set to 10000. MaxAgeSeconds specifies the time in seconds the browser can cache the response to a preflight (OPTIONS) request to a specific resource.

```
<?xml version="1.0" encoding="UTF-8"?>
   <CORSConfiguration>
     <CORSRule>
         <AllowedOrigin>www.aliyun.com</AllowedOrigin>
         <AllowedMethod>PUT</AllowedMethod>
         <MaxAgeSeconds>10000</MaxAgeSeconds>
     </CORSRule>
 </CORSConfiguration>
```

ii. Add a CORS rule for a bucket named examplebucket.

```
./ossutil64 cors --method put oss://examplebucket localfile.txt
```

If a similar output is displayed, the CORS rule is added for examplebucket.

```
0.299514(s) elapsed
```

Query CORS configurations for a bucket

- Command syntax

```
./ossutil64 cors --method get oss://bucketname [local_xml_file]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket for which you want to query CORS configurations.
local_xml_file	The name of the local file used to store the CORS configurations. Example: `localfile.txt`. If this parameter is not specified, the CORS configurations that are obtained are directly displayed.

- Examples

You can run the following command to query the CORS configurations of a bucket named examplebucket:

```
./ossutil64 cors --method get oss://examplebucket localfile.txt
```

If a similar output is displayed, the CORS configurations of examplebucket are obtained and stored in localfile.txt.

```
0.212407(s) elapsed
```

Delete CORS configurations for buckets

- Command syntax

```
./ossutil64 cors --method delete oss://bucketname
```

- Examples

Delete the CORS configurations of examplebucket.

```
./ossutil64 cors --method delete oss://examplebucket
```

If a similar output is displayed, the CORS configurations of examplebucket are deleted.

```
0.530750(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to add CORS configurations for a bucket named testbucket, which is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 cors --method put oss://testbucket localfile.txt -e oss-cn-hangzhou.aliyuncs.co
m -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the cors command, see Common options.

2.3.10. cors-options

Cross-origin resource sharing (CORS) is a standard cross-origin solution provided by HTML5 to allow web application servers to control cross-origin access. This way, the security of data transmission across origins is ensured. You can run the **cors-options** command to send HTTP OPTIONS requests to Object Storage Service (OSS) and check whether specific cross-origin requests are allowed.

> **Note**
>
> - In this topic, the sample command lines are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about how to configure CORS, see cors.

Command syntax

```
./ossutil64 cors-options oss://bucketname/[objectname]
[--acr-method <value>]
[--origin <value>]
[--acr-headers <value>]
```

The following table describes the parameters that you can configure when you run the cors-options command.

Parameter	Description
bucketname	The name of the bucket to access.

Parameter	Description
objectname	The name of the object to access.
--acr-method	The method that is allowed to use in the request. This parameter specifies the value of the `Access-Control-Request-Method` request header. Valid values: *GET*, *PUT*, *POST*, *DELETE*, and *HEAD*.
--origin	The origins from which you want to allow cross-origin requests. Example: `http://www.aliyun.com`.
--acr-headers	Specifies the value of the `Access-Control-Request-Headers` request header. This parameter specifies actual headers except for commonly used headers. You can specify multiple headers in a request. Separate multiple headers with commas (,). Example: `--acr-headers "header1,header2,header3"`.

Examples

You can run the following command to detect whether the examplebucket bucket allows a cross-origin request whose origin is `http://www.aliyun.com` and request method is `PUT`:

```
./ossutil64 cors-options --acr-method  put --origin "http://www.aliyun.com" oss://examplebucket
```

A similar output is displayed if examplebucket allows the cross-origin request.

```
Access-Control-Allow-Methods: GET, POST, PUT
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 0
0.079520(s) elapsed
```

A similar output is displayed if examplebucket denies the cross-origin request.

```
Error: oss: service returned error: StatusCode=403, ErrorCode=AccessForbidden, ErrorMessage="CORSResponse: This CORS request is not allowed. This is usually
 because the evalution of Origin, request method / Access-Control-Request-Method or Access-Control-Requet-Headers are not whitelisted by the resource's CORS
 spec.", RequestId=60F7F55F553DA2363138****
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

To detect whether a bucket named testbucket allows a cross-origin request whose origin is `http://www.alibabacloud.com` and request method is `PUT`, run the following command:

```
./ossutil64 cors-options --acr-method  put --origin "http://www.alibabacloud.com" oss:/test
bucket -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RV
AHUY9H****
```

For more information about other common options that you can use for the cors-options command, see Common options.

2.3.11. cp

2.3.11.1. Overview

The **cp** command is run to upload, download, or copy objects.

For more information about how to run this command to upload objects, see Upload objects.

For more information about how to run this command to download objects, see Download objects.

For more information about how to run this command to copy objects, see Copy objects.

2.3.11.2. Upload objects

This topic describes how to run the **cp** command to upload local files or directories to Object Storage Service (OSS).

Usage notes

- Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
- By default, when you use the **cp** command to upload an object, multipart upload and resumable upload are used. If the upload is interrupted before the object is completely uploaded, the uploaded data is stored as parts in an OSS bucket. To avoid additional storage fees, we recommend that you use the following methods to delete these parts if you no longer need them:
  - Manually delete parts. For more information, see Delete parts.
  - Configure lifecycle rules to automatically delete parts. For more information, see Configure lifecycle rules.

Command syntax

```
./ossutil64 cp file_url cloud_url
[-r, --recursive]
[-f --force]
[-u --update]
[--maxupspeed <value>]
[--enable-symlink-dir]
[--disable-all-symlink]
[--disable-ignore-error]
[--only-current-dir]
[--bigfile-threshold <value>]
[--part-size <value>]
[--checkpoint-dir <value>]
[--encoding-type <value>]
[--include <value>]
[--exclude <value>]
[--meta <value>]
[--acl <value>]
[--snapshot-path <value>]
[--disable-crc64]
[--disable-dir-object]
[--payer <value>]
[--tagging <value>]
[-j, --job <value>]
[--parallel <value>]
```

The following table describes the parameters that you can configure to run the cp command to upload objects to OSS.

Parameter	Description
file_url	Specifies the full path of the local file you want to upload. Examples: The path of the local file in Linux is `/localfolder/examplefile.txt`. The path of the local file in Windows is `D:\localfolder\examplefile.txt`.
cloud_url	Specifies the path of the OSS object. Format: `oss://bucketname/objectname`. Example: `oss://examplebucket/examplefile.txt`.
-r, --recursive	Specifies recursive operations. If you specify this option in a command, the command performs operations on all objects in a bucket that meet the specified condition. If you do not specify this option in a command, the command performs operations only on the specified object.
-f --force	Forces an operation without prompting the user for confirmation.
-u, --update	Specifies that ossutil uploads files from the source only when the objects do not exist in the destination or when the last modified time of the files is later than that of the objects in the destination bucket.
--maxupspeed	Specifies the maximum upload speed. Unit: KB/s. Default value: 0. The value of 0 indicates that the upload speed is not limited.

Parameter	Description
--enable-symlink-dir	Specifies that the subdirectory to which the symbolic link points is uploaded. By default, subdirectories are not uploaded.
--disable-all-symlink	Specifies that all objects in the subdirectory to which the symbolic link points and the subdirectory to which the symbolic link points are ignored during object upload.
--disable-ignore-error	Specifies that errors are not ignored during batch operations.
--only-current-dir	Specifies that only objects in the current directory are uploaded. Subdirectories in the current directory and objects in these subdirectories are not uploaded.
--bigfile-threshold	Specifies the maximum size of files that can be uploaded by using resumable upload. Unit: bytes. Default value: 104857600 (100 MB). Valid values: 0 to 9223372036854775807.
--part-size	Specifies the part size. Unit: bytes. By default, ossutil determines the part size based on the object size. Valid values: 1 to 9223372036854775807.
--checkpoint-dir	Specifies the directory in which the log information of resumable upload tasks is stored. Default value: `.ossutil_checkpoint`. When a resumable upload task fails, ossutil creates this directory and stores the checkpoint information about the task in this directory. ossutil deletes the directory after the task is completed. Therefore, if you specify a checkpoint directory, make sure that the directory can be deleted.
--encoding-type	Specifies the method used to encode the names of objects. Valid values: *url*. If you do not specify this parameter, the names of objects are not encoded.
--include	Specifies that the command applies to all objects that meet the specified conditions.
--exclude	Specifies that the command applies to all objects that do not meet the specified conditions.
--meta	Specifies the object metadata. The object metadata includes part of HTTP headers and user metadata that starts with `x-oss-meta-`. Format: `header:value#header:value`. Example: `Cache-Control:no-cache#Content-Encoding:gzip`. For more information about object metadata, see Manage object metadata.

Parameter	Description
--acl	Specifies the access control list (ACL) of an object. Default value: private. Valid values: • *default*: The ACL of the objects is the same as the ACL of the bucket in which the objects are stored. • *private*: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. • *public-read*: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. This may result in unrestricted access to the data in your bucket and unexpected high fees. Exercise caution when you set this parameter to this value. • *public-read-write*: All users, including anonymous users, can perform read and write operations on the objects in the bucket. This may result in unrestricted access to the data in your bucket and unexpected high fees. If a user uploads prohibited data or information, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set the object ACL to this value except in special cases.
--snapshot-path	Specifies the directory in which the snapshots of uploaded objects are stored. Next time the objects are uploaded while this option is specified, ossutil reads the snapshot information from the specified directory and performs an incremental upload.
--disable-crc64	Specifies that CRC-64 is disabled. By default, ossutil enables CRC-64 during data transmission.
--disable-dir-object	Specifies that no OSS objects are generated for the directory during object upload.
--payer	Specifies the payer of the traffic and request fees charged when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees, set this parameter to *requester*.
--tagging	Specifies the tags of objects in the following format: `TagkeyA=TagvalueA&TagkeyB=TagvalueB....`.
-j, --jobs	Specifies the number of files to upload concurrently. Valid values: 1 to 10000. Default value: 3.
--parallel	Specifies the number of concurrent tasks run to upload a single object. Valid values: 1 to 10000. By default, if you do not set this parameter, ossutil sets the value of this parameter based on the operation type and the object size.

The -j, --jobs and --parallel options in the preceding command syntax can be used to adjust performance if the default number of concurrent tasks does not meet your performance requirements. By default, ossutil calculates the number of concurrent operations based on the object size. When you upload multiple large objects, the actual number of concurrent tasks is calculated by multiplying the number of jobs by the number of concurrent operations.

- We recommend that you adjust the number of concurrent tasks to a value smaller than 100 if the resources such as network bandwidth, memory, and CPU of your ECS instance or server are limited. If resources such as the network bandwidth, memory, and CPU are available, you can increase the number of concurrent operations.
- If the number of concurrent tasks is too large, the upload performance of ossutil may decrease or an end-of-file (EOF) error may occur because thread resources are switched and threads compete for resources. To resolve this issue, you must adjust the values of the -j, --jobs and --parallel options based on the actual conditions of machines. To perform stress testing, set a small value for the two options before you incrementally increase them to the optimal values.

Sample environment

In this topic, local files or directories are uploaded from a Linux system to OSS. You can modify the parameters in the examples based on your operating system and environment. This topic uses the following common examples:

- Local file: examplefile.txt (a file in the root directory)
- Local directory: localfolder (a directory in the root directory)
- OSS bucket: examplebucket
- Specified directory in the OSS bucket: desfolder

Simple upload

You can use ossutil to upload local files to OSS. The following examples show how to run the cp command to upload objects to OSS by using simple upload:

- Upload a single local file

  If you do not specify the name of the uploaded object, the name of the local file is used as the object name. If you specify the name of the uploaded object, the object is stored in OSS based on the specified name.

  ```
  ./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/
  ```

- Upload the files in a local directory

  You can add the -r parameter to the **cp** command to upload files from a local directory to a specified path of OSS.

  ```
  ./ossutil64 cp -r localfolder/ oss://examplebucket/desfolder/
  ```

- Upload a local directory and the files inside

  You can add the -r option to the **cp** command and add the name of a local directory to upload the local directory and the files inside to a specified path of OSS.

  ```
  ./ossutil64 cp -r localfolder/ oss://examplebucket/desfolder/localfolder/
  ```

- Upload a single file and specify the --meta parameter

When you run the cp command to upload a local file, you can add the --meta option to the command and configure the metadata of the file in the following format: `header:value#header:value...` .

```
./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/examplefile.txt --meta=Cache-Control:no-cache#Content-Encoding:gzip
```

- Upload a directory without uploading existing objects

  When you resume a failed batch upload task, you can specify the --update (abbreviated as the -u) option to skip uploaded files. This way, you can upload only incremental data to OSS.

  ```
  ./ossutil64 cp -r localfolder/ oss://examplebucket/desfolder/ -u
  ```

- Upload objects to a bucket for which pay-by-requester is enabled

  ```
  ./ossutil64 cp localfolder/examplefile.txt oss://examplebucket/ --payer=requester
  ```

- Upload objects only in the current directory

  ```
  ./ossutil64 cp localfolder/ oss://examplebucket/desfolder/ --only-current-dir -r
  ```

- Upload a directory without generating an object for the uploaded directory

  In OSS, a directory is an object that is 0 KB in size and has a name that ends with a forward slash (/). If you specify the --disable-dir-object parameter in the cp command to upload a directory, OSS does not generate an object for the uploaded directory. However, you can view the directory in the OSS console. If you delete all objects from the directory, the directory is also deleted.

  ```
  ./ossutil64 cp localfolder/ oss://examplebucket/desfolder/ --disable-dir-object -r
  ```

- Upload objects including those in subdirectories to which symbolic links point

  ```
  ./ossutil64 cp localfolder/ oss://examplebucket/desfolder/ --enable-symlink-dir -r
  ```

- Upload objects and ignore the objects in subdirectories and subdirectories to which symbolic links point

  ```
  ./ossutil64 cp localfolder/ oss://examplebucket/desfolder/ -r --disable-all-symlink
  ```

Configure the maximum upload speed

When you upload an object, you can set --maxupspeed to limit the maximum upload speed. Unit: KB/s. Examples:

- Upload an object to OSS and set the maximum upload speed to 1 MB/s

  ```
  ./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/ --maxupspeed 1024
  ```

- Upload a directory to OSS and set the maximum upload speed to 1 MB/s

  ```
  ./ossutil64 cp -r localfolder/ oss://examplebucket/desfolder/ --maxupspeed 1024
  ```

Configure tagging for an object when you upload the object

When you run the cp command to upload an object, you can specify the --tagging option to configure tags for the object. Separate multiple tags with ampersands (&). Example:

```
./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/ --tagging "abc=1&bcd=2&……"
```

For more information about object tagging, see object-tagging (add, modify, query, and delete object tags).

Specify the storage class of an object when you upload the object

When you run the cp command to upload a local file, you can specify the --meta option to specify the storage class of the object. OSS supports the following storage classes:

- Standard
- Infrequent Access (IA)
- Archive

If you do not specify the storage class in the command, the storage class of the uploaded object is the same as that of the bucket in which the object is stored. For more information, see Overview. Examples:

- Upload an object and set the storage class of the object to IA

  ```
  ./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/ --meta X-oss-Storage-Class:IA
  ```

- Upload a directory and set the storage class of all objects in the specified directory to Standard

  ```
  ./ossutil64 cp localfolder/ oss://examplebucket/desfolder/ --meta X-oss-Storage-Class:Standard -r
  ```

Specify the ACL of an object when you upload the object

When you run the cp command to upload objects, you can specify the --meta option to set the access control list (ACL) of the objects. OSS supports the following object ACLs:

- default: The ACL of an object is the same as that of the bucket in which the object is stored.
- private
- public-read
- public-read-write

The following examples show how to specify the ACL of objects when you run the cp command to upload objects:

- Upload an object and set the ACL of the object to private

  ```
  ./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/ --meta x-oss-object-acl:private
  ```

- Upload an object and set the ACL of the object to public read

  ```
  ./ossutil64 cp localfolder/ oss://examplebucket/desfolder/ --meta x-oss-object-acl:public-read  -r
  ```

Specify an encryption method for an object when you upload the object

When you upload an object, you can specify the server-side encryption method for the object. The object is stored in the specified bucket after the object is encrypted. Examples:

- Upload an object and set the encryption method of the object to SSE-OSS and the encryption algorithm to AES-256

```
./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/ --meta=x-oss-server-side-encryption:AES256
```

- Upload an object and set the encryption method of the object to SSE-KMS without specifying a customer master key (CMK) ID

```
./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/ --meta=x-oss-server-side-encryption:KMS
```

  When you set the encryption method to SSE-KMS, you are charged for the usage of CMK IDs. For more information, see KMS billing methods.

- Upload an object, set the encryption method of the object to SSE-KMS, and then specify a CMK ID

```
./ossutil64 cp examplefile.txt oss://examplebucket/desfolder/ --meta=x-oss-server-side-encryption:KMS#x-oss-server-side-encryption-key-id:7bd6e2fe-cd0e-483e-acb0-f4b9e1******
```

For more information about server-side encryption, see Server-side encryption.

Generate snapshots for objects when you upload the objects

When you run the cp command to batch upload objects and specify the --snapshot-path option in the command, ossutil creates the snapshots of the uploaded objects in the specified directory to record the last modified time of the objects. In subsequent upload tasks, ossutil determines whether to skip existing objects based on their last modified time. If you want to specify the --snapshot-path option in the command, make sure that the objects in OSS are not modified by other users since the last time when the objects are uploaded. The --snapshot-path option is used to accelerate incremental batch uploads. The following example shows how to generate snapshots for objects when you run the cp command to upload the objects:

```
./ossutil64 cp -r localfolder/ oss://examplebucket/desfolder/ --snapshot-path=path
```

Notice

- ossutil does not delete snapshots stored in the directory specified by the *snapshot-path* option. To prevent snapshots from consuming too much storage space, delete snapshots that you no longer need from the directory specified by the *snapshot-path* option.
- Additional overheads are required to read and write snapshot information. We recommend that you do not use this option in the following scenarios: The number of objects to upload is small. Network conditions are good. Other users need to perform operations on those objects. To implement incremental upload, we recommend that you use the --update option.
- You can specify both the --update and --snapshot-path options in a command. ossutil determines whether to skip an object first based on the snapshots stored in the directory specified by --snapshot-path. If no snapshots are generated for the object, ossutil determines whether to skip the file based on the --update option.

Batch upload objects that meet specified conditions

When you run the cp command to batch upload objects and specify the --include and --exclude options in the command, ossutil batch uploads objects that meet the specified conditions.

The --include and --exclude options support the following formats:

- Asterisk (*): matches all characters. For example, **.txt* indicates all *TXT* files.
- Question mark (?): matches a single character. For example, *abc?.jpg* specifies all JPG objects whose names start with abc and are followed by a single character such as *abc1.jpg*.
- [sequence]: matches characters in a sequence. For example, *abc[1-5].jpg* specifies the objects whose names start with abc and are followed by a number contained in sequence [1-5]. The objects include *abc1.jpg, abc2.jpg, abc3.jpg, abc4.jpg, and abc5.jpg*.
- [sequence]: matches characters in a sequence. For example, *abc[!0-7].jpg* specifies that objects *abc0.jpg, abc1.jpg, abc2.jpg, abc3.jpg, abc4.jpg, and abc5.jpg, abc6.jpg, and abc7.jpg* are not matched.

A rule can contain multiple conditions specified by --include and --exclude. After these conditions are configured, ossutil reads each rule from left to right to obtain the final matching results. If the *test.txt* object exists in a directory for which conditions are specified, results are generated based on different matching rules.

- Rule 1: `--include "*test*" --exclude "*.txt"`. When ossutil reads the `--include "*test*"` condition, the *test.txt* object matches the condition. When ossutil reads the `--exclude "*.txt"` condition, the *test.txt* object is excluded because the object name is in the TXT format. The final matching results exclude the *test.txt* object.
- Rule 2: `--exclude "*.txt" --include "*test*"`. When ossutil reads the `--exclude "*.txt"` condition, the *test.txt* object is excluded. When ossutil reads the `--include "*test*"` condition, the *test.txt* object matches the condition because its name contains test. The final matching results include the *test.txt* object.
- Rule 3: `--include "*test*" --exclude "*.txt" --include "te?t.txt"`. When ossutil reads the `--include "*test*"` condition, the *test.txt* object matches the condition. When ossutil reads the `--exclude "*.txt" condition`, the *test.txt* object is excluded because the object name is in the TXT format. When ossutil reads the `--include "te?t.txt"` condition, the *test.txt* object matches the condition. The final matching results include the *test.txt* object.

> **Notice** Conditions that include directory names such as `--include "/usr/test/.jpg"` are not supported.

The following examples show how to run the cp command to specify conditions to upload only objects that match the conditions:

- Upload all objects that are in the *TXT* format

```
./ossutil64 cp localfolder/ oss://examplebucket/desfolder/ --include "*.txt" -r
```

- Upload all objects that contain *abc* in their names and are not in the *JPG* or *TXT* format

```
./ossutil64 cp localfolder/ oss://examplebucket/desfolder/ --include "*abc*" --exclude "*
.jpg" --exclude "*.txt" -r
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of a specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of a specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to upload the exampleobject.txt local file to the destfolder directory of a bucket named examplebucket. The bucket is located in the China (Shanghai) region and owned by another Alibaba Cloud account.

```
./ossutil64 cp exampleobject.txt oss://examplebucket/desfolder/ -e oss-cn-shanghai.aliyuncs
.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the cp command, see Common options.

2.3.11.3. Download objects

This topic describes how to run the **cp** command to download objects from Object Storage Service (OSS) to your local computer.

> **Note** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 cp cloud_url file_url
[-r, --recursive]
[-f --force]
[-u --update]
[--maxdownspeed <value>]
[--disable-ignore-error]
[--only-current-dir]
[--bigfile-threshold <value>]
[--part-size <value>]
[--checkpoint-dir <value>]
[--range <value>]
[--encoding-type <value>]
[--include <value>]
[--exclude <value>]
[--meta <value>]
[--acl <value>]
[--snapshot-path <value>]
[--disable-crc64]
[--payer <value>]
[--partition-download <value>]
[-j, --job <value>]
[--parallel <value>]
[--version-id <value>]
```

The following table describes the parameters that you can configure when you run this command to download objects from OSS.

Parameter	Description
cloud_url	Specifies the path of the object that you want to download from OSS. Format: `oss://bucketname/objectname`. Example: `oss://examplebucket/examplefile.txt`.
file_url	Specifies the local path to which you want to download the object from OSS. Examples: The local path in Linux is `/localfolder/examplefile.txt`. The local path in Windows is `D:\localfolder\examplefile.txt`.
-r, --recursive	Specifies recursive operations. If this option is specified, commands that support this option are run to perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, commands that support this option are run to perform operations only on the specified object.
-f --force	Forces an operation without prompting the user for confirmation.
-u, --update	Specifies that ossutil downloads an object from OSS only when the destination file does not exist in the local computer or when the last modified time of the source object is later than that of the destination file.
--maxdownspeed	Specifies the maximum download speed. Unit: KB/s. The default value is 0, which indicates that the download speed is unlimited.
--disable-ignore-error	Specifies that errors are not ignored during batch operations.
--only-current-dir	Specifies that only objects in the current directory are downloaded. Subdirectories in the current directory and objects in these subdirectories are not downloaded.
-bigfile-threshold	Specifies the maximum size of objects that can be downloaded by using resumable download. Unit: bytes. Default value: 104857600 (100 MB). Valid values: 0 to 9223372036854775807.
--part-size	Specifies the part size. Unit: bytes. By default, ossutil determines the part size based on the object size. Valid values: 1 to 9223372036854775807.
--checkpoint-dir	Specifies the directory in which the log information of resumable upload or download tasks is stored. Default value: `.ossutil_checkpoint`. When a resumable download task fails, ossutil creates this directory and stores the checkpoint information about the task in this directory. ossutil deletes the directory after the task is completed. Therefore, if you specify a checkpoint directory, make sure that ossutil can delete the directory.

Parameter	Description
--range	Specifies that a specific range of the object content is downloaded and stored as a new object in the destination local path. The minimum start value of the range is 0, which indicates the byte 0 of the content of the object. You can set this parameter in one of the following formats: • Data range For example, a value of `3-9` indicates a range from byte 3 to byte 9, which includes byte 3 and byte 9. • Start value For example, a value of `3-` indicates a range from byte 3 to the end of the object, which includes byte 3. • End value For example, the value of `-9` indicates a range from byte 0 to byte 9, which includes byte 9.
--encoding-type	Specifies the method used to encode the names of objects. Valid values: *url*. If you do not specify this parameter, the names of objects are not encoded.
--include	Specifies that the command applies to all objects that meet the specified conditions.
--exclude	Specifies that the command applies to all objects that do not meet the specified conditions.
--meta	Specifies that the metadata of an object is in the `header:value#header:value` format. Example: `Cache-Control:no-cache#Content-Encoding:gzip`. For more information about object metadata, see set-meta.

Parameter	Description
--acl	Specifies the access control list (ACL) of the object to download. Default value: private. Valid values: • *default*: The ACL of the objects is the same as the ACL of the bucket in which the objects are stored. • *private*: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. • *public-read*: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. This may result in unexpected access to the data in your bucket and unexpectedly high fees. Exercise caution when you set this parameter to public-read. • *public-read-write*: All users, including anonymous users, can perform read and write operations on the objects in the bucket. This may result in unexpected access to the data in your bucket and unexpectedly high fees. If a user uploads prohibited data or information, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set your bucket ACL to public read/write except in special cases.
--snapshot-path	Specifies the directory in which the snapshots of downloaded objects are stored. In the next download task, ossutil reads the snapshots in this directory to download only incremental objects.
--disable-crc64	Specifies that CRC-64 is disabled. By default, CRC-64 is enabled when you use ossutil to transfer data.
--payer	The payer of the traffic and request fees charged when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees charged during downloads, set this parameter to *requester*.
--partition-download	Specifies the partition in which the object that you want to download is stored. The value of this option is in the `partition number: the total number of partitions` format. You can run multiple ossutil commands on different machines at the same time. A value of 1:5 indicates that the current command downloads partition 1 and the object is split into five partitions. Partitions are numbered from 1. Partitioning rules for objects are determined by ossutil. This option splits an object into multiple partitions that can be concurrently downloaded by multiple ossutil commands. Each ossutil command downloads its own partition.
-j, --job	Specifies the number of objects to download concurrently. Valid values: 1 to 10000. Default value: 3.
--parallel	Specifies the number of concurrent tasks run to download a single object. Valid values: 1 to 10000. By default, if you do not set this parameter, ossutil sets the value of this parameter based on the operation type and the size of the object to download.

Parameter	Description
--version-id	Specifies the version ID of the object that you want to download. You can use this option for objects only in a bucket that has versioning enabled.

When the default configuration of the number of concurrent tasks based on the -j, --jobs and --parallel options cannot meet your performance requirements, you can configure these two parameters based on your requirements. By default, ossutil calculates the number of concurrent operations based on the size of the object to download. When you download multiple large objects, the actual number of concurrent tasks is calculated by multiplying the value of -j, --jobs by the value of --parallel.

- We recommend that you adjust the number of concurrent tasks to a value smaller than 100 if the resources such as network bandwidth, memory, and CPU of your Elastic Compute Service (ECS) instance or server are limited. If the maximum resources such as the network bandwidth, memory, and CPU are not reached, you can increase the number of concurrent operations.
- If the number of concurrent tasks is too large, the download performance of ossutil may decrease, or an end-of-file (EOF) error may occur because thread resources are switched and threads compete for resources. To resolve this issue, you must adjust the values of the -j, --jobs and --parallel options based on the actual conditions of machines. When you perform stress testing, set a small value for the two options before you incrementally increase them to the optimal values.

Sample environment

In this topic, objects are downloaded from OSS to your local directory in Linux. You can modify the parameters in the following examples based on your operating system and environment. Sample environment:

- Operating system: Linux
- Destination bucket: examplebucket
- Directory of the destination bucket: desfolder
- Local file: examplefile.txt. This file is stored in the root directory of the local computer.
- Local directory: localfolder. This directory is stored in the root directory of the local computer.

Download a single object

If you do not specify a name for the destination object, the name of the source object is used by default. If you specify a name for the destination object, the specified name is used in the local computer.

- You can run the following command to save the destination object with the name of the source object:

```
./ossutil64 cp oss://examplebucket/destfolder/examplefile.txt localfolder/
```

- You can run the following command to save the destination object with a specified name:

```
./ossutil64 cp oss://examplebucket/destfolder/examplefile.txt localfolder/example.txt
```

Download multiple objects

You cannot use ossutil to download multiple objects at the same time by specifying multiple object names. You can download multiple objects at the same time by using the following methods:

- Download a type of objects

  If you want to download multiple objects whose names contain the same prefix or suffix, you can use the *--include* and *--exclude* options to download objects that meet specific conditions.

  - You can run the following command to download all objects that are not in the *JPG* format:

    ```
    ./ossutil64 cp oss://examplebucket/destfolder/ localfolder/ --exclude "*.jpg" -r
    ```

  - You can run the following command to download all objects that contain *abc* in their names and are not in the *JPG* or *TXT* format:

    ```
    ./ossutil64 cp oss://examplebucket/destfolder/ localfolder/ --include "*abc*" --exclude
    "*.jpg" --exclude "*.txt" -r
    ```

- Download a directory and its subdirectories

  You can run the following command to download all objects in a directory, including subdirectories in the directory:

  ```
  ./ossutil64 cp -r oss://examplebucket/destfolder/ localfolder/
  ```

  When a batch download task fails or when you want to implement incremental download, you can use the *--update* option that can be shortened to -u to skip objects that have been downloaded. If an object does not have the same name as local files, or the object is last modified later than the local file that has the same name, ossutil downloads the object. Otherwise, ossutil skips the object. Example:

  ```
  ./ossutil64 cp -r oss://examplebucket/destfolder/  localfolder/  --update
  ```

- Download a directory (excluding its subdirectories)

  If you want to download only the current directory and ignore its subdirectories, use the *--only-current-dir* option. Example:

  ```
  ./ossutil64 cp oss://examplebucket/destfolder/ localfolder/ --only-current-dir -r
  ```

Limit download speed

To limit the maximum download speed, you can use the --maxdownspeed option. Unit: KB/s. Examples:

- You can run the following command to download objects from OSS and set the value of the --maxdownspeed parameter to 1 MB/s:

  ```
  ./ossutil64 cp oss://examplebucket/destfolder/examplefile.txt  localfolder/ --maxdownspee
  d 1024
  ```

- You can run the following command to download directories from OSS and set the value of the --maxdownspeed parameter to 1 MB/s:

  ```
  ./ossutil64 cp -r  oss://examplebucket/destfolder/  localfolder/ --maxdownspeed 1024
  ```

Download a range of an object

To download a range of the content of an object, use the --range option. For example, you can run the following command to download the 10th to 20th characters of the *examplefile.txt* object to your local computer as an object:

```
./ossutil64 cp oss://examplebucket/destfolder/examplefile.txt localfolder/  --range=10-20
Succeed: Total num: 1, size: 11. OK num: 1(download 1 objects).
```

Download a directory and generate snapshot information

If you specify the --snapshot-path option when you download multiple objects, ossutil takes a snapshot of the objects and stores the snapshot information in a specified directory. Next time the objects are downloaded while this option is specified, ossutil reads the snapshot information from the specified directory and downloads only incremental objects. For more information, see Generate snapshots for objects when you upload the objects.

```
./ossutil64 cp -r oss://examplebucket/destfolder/ localfolder/ --snapshot-path=path
```

Download a specified version of an object from a versioned bucket

After you enable versioning for a bucket, objects that are overwritten or deleted in the bucket are saved as previous versions. You can run the following **cp** command to which the --version-id parameter is added to download a specified version of an object:

```
./ossutil64 cp oss://my-bucket/test.jpg localfolder/ --version-id  CAEQARiBgID8rumR2hYiIGUy
OTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3MDRk
```

> **Notice**
>
> - To use the --version-id option, you must run the ls --all-versions command to query the version IDs of the object.
> - The --version-id option can be used for objects only in versioned buckets. For more information about the command run to enable versioning for a bucket, see bucket-versioning.

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of a specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of a specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to download the exampleobject.txt object in the root directory of the examplebucket bucket to the local directory named localfolder. The source bucket is located in the China (Shanghai) region and is owned by another Alibaba Cloud account.

```
./ossutil64 cp oss://examplebucket/exampleobject.txt  localfolder/  -e oss-cn-shanghai.aliy
uncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the cp command, see Common options.

2.3.11.4. Copy objects

This topic describes how to run the **cp** command to copy an Object Storage Service (OSS) object to other directories within the same bucket or to other buckets that are located in the same region as the source object. Currently, only objects can be copied. Parts that are generated by incomplete multipart copy tasks cannot be copied.

> **Note** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Syntax

```
./ossutil64 cp cloud_url cloud_url
[-r, --recursive]
[-f --force]
[-u --update]
[--disable-ignore-error]
[--only-current-dir]
[--bigfile-threshold <value>]
[--part-size <value>]
[--checkpoint-dir <value>]
[--encoding-type <value>]
[--include <value>]
[--exclude <value>]
[--meta <value>]
[--acl <value>]
[--disable-crc64]
[--payer <value>]
[-j, --job <value>]
[--parallel <value>]
[--version-id <value>]
```

The following table describes the parameters that you can configure when you run this command to download objects from OSS.

Parameter	Description
cloud_url	Specifies the paths of the source object and the destination object. Format: `oss://bucketname/objectname`. For example, to copy the srcobject.jpg source object from the examplebucket bucket to the destobject.jpg object in the same bucket, specify the path of the source object as `oss://examplebucket/srcobject.jpg` and the path of the destination bucket as `oss://examplebucket/destobject.jpg`.
-r, --recursive	Specifies recursive operations. If this option is specified, commands that support this option are run to perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, commands that support this option are run to perform operations only on the specified object.
-f --force	Specifies the operation to forcibly perform without prompting the user for confirmation.

Parameter	Description
-u, --update	Specifies that ossutil copies the source object only when the object does not exist in the destination bucket or when the last modified time of the source object is later than that of the destination object.
--disable-ignore-error	Specifies that errors are not ignored during batch operations.
--only-current-dir	Specifies that only objects in the current directory are copied. Subdirectories in the current directory and objects in these subdirectories are not copied.
-bigfile-threshold	Specifies the maximum size of files that can be synchronized by using resumable upload or download. Unit: bytes. Default value: 104857600 (100 MB). Valid values: 0 to 9223372036854775807.
--part-size	Specifies the part size. Unit: bytes. By default, ossutil determines the part size based on the object size. Valid values: 1 to 9223372036854775807.
--checkpoint-dir	Specifies the directory in which the log information of resumable upload or download tasks is stored. Default value: `.ossutil_checkpoint`. When a resumable synchronization task fails, ossutil creates this directory and stores the checkpoint information about the task in this directory. ossutil deletes the directory after the task is completed. Therefore, if you specify a checkpoint directory, make sure that the directory can be deleted.
--encoding-type	Specifies the method used to encode the names of objects. Valid values: *url*. If you do not specify this parameter, the names of objects are not encoded.
--include	Specifies that the command applies to all files that meet the specified conditions.
--exclude	Specifies that the command applies to all files that do not meet the specified conditions.
--meta	Specifies that the metadata of an object is in the `header:value#header:value` format. Example: `Cache-Control:no-cache#Content-Encoding:gzip`. For more information about object metadata, see set-meta.

Parameter	Description
--acl	Specifies the access control list (ACL) of an object. Default value: private. Valid values: • *default*: The ACL of the objects is the same as the ACL of the bucket in which the objects are stored. • *private*: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. • *public-read*: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. This may result in unexpected access to the data in your bucket and unexpectedly high fees. If a user uploads prohibited data or information, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set the object ACL to this value except in special cases. • *public-read-write*: All users, including anonymous users, can perform read and write operations on the objects in the bucket. This may result in unexpected access to the data in your bucket and unexpectedly high fees. Exercise caution when you set this parameter to this value.
--disable-crc64	Specifies that CRC-64 is disabled. By default, CRC-64 is enabled when you use ossutil to transfer data.
--payer	The payer of the traffic and request fees charged when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees charged during the copy operation, set this parameter to *requester*.
-j, --jobs	Specifies the number of files to synchronize concurrently. Valid values: 1 to 10000. Default value: 3.
--parallel	Specifies the number of concurrent tasks run to synchronize a single file. Valid values: 1 to 10000. By default, if you do not set this parameter, ossutil sets the value of this parameter based on the operation type and the size of the object to download.
--version-id	Specifies the version ID of the object that you want to copy. You can use this option for objects only in a bucket that has versioning enabled.

When the default configuration of the number of concurrent tasks based on the -j, --jobs and --parallel parameters cannot meet your requirements, you can configure these two parameters based on your requirements. By default, ossutil calculates the number of concurrent tasks based on the size of the object. When you copy multiple large objects, the actual number of concurrent tasks is calculated by multiplying the number of concurrent jobs by the number of concurrent operations.

- We recommend that you adjust the number of concurrent tasks to a value smaller than 100 if the resources such as network bandwidth, memory, and CPU of your Elastic Compute Service (ECS) instance or server are limited. If resources such as the network bandwidth, memory, and CPU are available, you can increase the number of concurrent tasks.
- If the number of concurrent tasks is too large, the copy performance of ossutil may decrease, or an

end-of-file (EOF) error may occur because thread resources are switched and threads compete for resources. To resolve this issue, you must adjust the values of the -j, --jobs and --parallel options based on the actual conditions of machines. When you perform stress testing, set a small value for the two options before you incrementally increase them to the optimal values.

Sample environment

In this topic, objects are copied between different directories or buckets in Linux. You can modify the parameters in the following examples based on your operating system and environment. Sample environment:

- Operating system: Linux
- Source bucket: examplebucket1
- Directory 1 specified by the source bucket: srcfolder1
- Directory 2 specified by the source bucket: srcfolder2
- Source object: examplefile.txt
- Destination bucket: examplebucket2
- Directory specified by the destination bucket: desfolder

Example for simple copy

- Copy a single object

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket1/srcfolder2/
```

- Copy a directory

  You can run the following command in which the -r parameter is added to copy multiple objects:

```
./ossutil64 cp oss://examplebucket1/srcfolder1/ oss://examplebucket2/desfolder/ -r
```

- Copy incremental objects

  To copy multiple objects, if you specify the --update parameter, ossutil copies the objects only when the destination objects do not exist, or the last modified time of the source objects is later than that of the destination objects. You can run the following command to copy multiple objects:

```
./ossutil64 cp oss://examplebucket1/srcfolder1/ oss://examplebucket2/path2/ -r --update
```

  This option can be used to copy objects that fail to copy or to skip copied objects when you copy incremental objects.

- Rename the object

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket1/example.txt
```

  When you run the **cp** command to rename an object, the original object still exists. You can delete the original object after you rename the object.

- Copy only objects in the current directory and ignore subdirectories

```
./ossutil64 cp oss://examplebucket1/srcfolder1/ oss://examplebucket1/srcfolder2/ --only-c
urrent-dir -r
```

- Modify the metadata of the object

When you copy an object, you can use the --meta parameter to modify the object metadata in the `header:value#header:value...` format.

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket1/ --meta=Cache-Control:no-cache
```

- Copy an object from a bucket that has pay-by-requester enabled to a bucket that has pay-by-requester disabled

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket2/desfolder/  --payer=requester
```

Modify the storage class of an object

When you overwrite an object, you can use the --meta option to convert the storage class of the object. OSS supports the following storage classes:

- Standard
- Infrequent Access (IA)
- Archive

For more information, see Overview. Examples:

- You can run the following command to convert the storage class of the specified object to Archive:

```
./ossutil64 cp oss://examplebucket1/srcfolder1/examplefile.txt oss://examplebucket1/srcfolder1/examplefile.txt --meta X-oss-Storage-Class:Archive
```

- You can run the following command to convert the storage class of all objects in a specified directory to Standard:

```
./ossutil64 cp oss://examplebucket1/srcfolder1/ oss://examplebucket1/srcfolder1/ --meta X-oss-Storage-Class:Standard -r
```

> **Notice**
>
> - You cannot convert the storage class of an Archive or Clod Archive object to other storage classes by running the **cp** command. You must first run the restore (restore objects) command to restore the object, and then run the **cp** command to convert the storage class of the object.
> - When you run the **cp** command to overwrite an object, fees may be charged. If an object of the IA, Archive, or Cold Archive storage class is stored for less than the minimum storage duration and is overwritten, you are charged for the minimum storage duration, which includes the remaining duration. For more information, see Storage fees.

Modify the tags of an object

You can run the following command to which the --tagging parameter is added to modify the tags of an object. Separate the tags with ampersands (&). In this case, run the following command:

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket1/ --tagging "abc=1&bcd=2&……"
```

For more information about object tagging, see object-tagging (add, modify, query, and delete object tags).

Copy and encrypt an object

You can specify the server-side encryption method when you copy an object and store the encrypted object in a bucket. For more information about server-side encryption, see Server-side encryption.

- You can run the following command to copy an object and set the server-side encryption method to AES-256:

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket1/srcfolder2/ --me
ta=x-oss-server-side-encryption:AES256
```

- You can run the following command to copy an object and set the server-side encryption method to KMS:

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket2/desfolder/ --met
a=x-oss-server-side-encryption:KMS
```

> **Notice** When you use KMS to encrypt an object, OSS creates a customer master key (CMK) in the KMS console for the object and charges a small amount of fees when the KMS API operation is called. For more information, see Billing.

- You can run the following command to copy an object, set the encryption method of the object to SSE-KMS, and specify a CMK ID:

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket2/desfolder/ --met
a=x-oss-server-side-encryption:KMS#x-oss-server-side-encryption-key-id:7bd6e2fe-cd0e-483e
-acb0-f4b9e1******
```

Recover objects in a versioning-enabled bucket

After you enable versioning for a bucket, objects that are overwritten or deleted in the bucket are saved as previous versions. You can run the following **cp** command to which the --version-id parameter is added to overwrite the previous version of an object as the latest version: In this case, run the following command:

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket2/ --version-id  CAE
QARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3MDRk
```

> **Notice**
>
> - To use the --version-id parameter, you must run the ls --all-versions command to query version IDs of the object.
> - The --version-id parameter can be used for objects only in versioning-enabled buckets. For more information about the command used to enable versioning for a bucket, see bucket-versioning.

Copy multiple objects that meet specified conditions

To copy objects that match specified conditions, you can use the --include and --exclude parameters. For more information, see Batch upload objects that meet specified conditions.

- You can run the following command to copy all objects that are not in the *JPG* format:

```
./ossutil64 cp oss://examplebucket1/srcfolder1/ oss://examplebucket2/desfolder/ --exclude
"*.jpg" -r
```

- You can run the following command to copy all objects that contain *abc* in their names and are not in the *JPG* or *TXT* format:

```
./ossutil64 cp oss://examplebucket1/srcfolder1/ oss://examplebucket2/desfolder/ --include
"*abc*" --exclude "*.jpg" --exclude "*.txt" -r
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of a specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of a specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to copy the srcobject.png object in the root directory of the examplebucket bucket to the destination bucket named destbucket. The source bucket is located in the China (Shanghai) region and is owned by another Alibaba Cloud account.

```
./ossutil64 cp oss://examplebucket/srcobject.png  oss://destbucket  -e oss-cn-shanghai.aliy
uncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the cp command, see Common options.

2.3.12. create-symlink (create a symbolic link)

Symbolic links can be used to access objects that are commonly used in buckets. The **create-symlink** command is run to create a symbolic link. After you create a symbolic link for an object, you can use the symbolic link to access the object. Symbolic links work in a similar manner to shortcuts in Windows.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 create-symlink cloud_url target_object [--encoding-type <value>] [--payer <valu
e>]
```

The following table describes the parameters that you can configure in this command.

Parameter	Description
cloud_url	The full path of the symbolic link you want to create.
target_object	The full path of the object to which the symbolic link points. A symbolic link and the object to which the symbolic link points must be in the same bucket.

Parameter	Description
--encoding-type	The method used to encode the object names specified in `cloud_url` and `target_object`. Valid value: *url*. If you do not specify this parameter, the object names are not encoded.
--payer	The payer of the traffic and request fees incurred when the command is run. If you want that the requester who accesses the resources in the specified path charged for the traffic and request fees incurred when the command is run, set this parameter to *requester*.

Examples

When you run this command to create a symbolic link, ossutil does not check whether the object to which the symbolic link points to exists. If the object exists, the created symbolic link can access the object. If the object does not exist, the created symbolic link cannot access the object. To determine whether the object to which a symbolic link points exists, run the ls command to query all objects in the bucket.

The following examples show how to create a symbolic link that points to an existing object.

> **Notice** If the name of the symbolic link you want to create is the same as that of an existing symbolic link, the existing symbolic link is overwritten.

- Create a symbolic link named test.jpg in the root directory of a bucket named examplebucket and point the symbolic link to an object named exampleobject.jpg in the root directory of examplebucket.

  ```
  ./ossutil64 create-symlink  oss://examplebucket/test.jpg  oss://examplebucket/exampleobje
  ct.jpg
  ```

- Create a symbolic link named example.jpg in the destfolder directory of a bucket named examplebucket and point the symbolic link to an object named test.jpg in the root directory of examplebucket. Specify that all fees incurred when the command is run are paid by the requester.

  ```
  ./ossutil64 create-symlink  oss://examplebucket/destfolder/example.jpg  oss://examplebuck
  et/test.jpg --payer requester
  ```

- If a similar output is displayed, the symbolic link is created.

  ```
  0.106744(s) elapsed
  ```

  After you create a symbolic link, you can run the read-symlink or stat command to query the information about the symbolic link, such as the ETag value and last update time.

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by multiple Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to create a symbolic link named testobject.png that points to an object named exampleobject.png in a bucket named testbucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 create-symlink  oss://testbucket/testobject.png  oss://testbucket/exampleobject
.png -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAH
UY9H****
```

For more information about other common options, see Common options.

2.3.13. du (query object sizes)

The **du** command is used to obtain the total size of all objects within a bucket or directory.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 du oss://bucketname[/prefix] [--payer requester] [--all-versions][--block-size
<value>]
```

The following table describes the parameters that you can configure in the du command.

Parameter	Description
bucketname	The name of the bucket in which the objects whose total size you want to query are stored.
prefix	The path of a directory in which the objects whose total size you want to query are stored, or a prefix that is contained in the names of all objects whose size you want to query.
--payer	The payer of the traffic and request fees incurred during queries. If you want that the traffic and request fees incurred during queries are paid by the requester who accesses the resources in the specified path, set this parameter to *requester*.
--all-versions	Specifies whether to query the total size of all versions of objects. By default, if you do not specify this parameter in the command, the total size of only the current versions of objects is queried.
--block-size	The unit of the obtained total size of objects within the specified bucket or directory. Valid values: *KB*, *MB*, *GB*, and *TB*. By default, if you do not specify this parameter in the command, the obtained total size of objects is measured in bytes. > **Notice** This parameter applies to ossutil 1.7.3 and later versions.

Query the total size of all versions of objects in the specified bucket

You can run the following command to query the total size of all versions of objects in a bucket named examplebucket:

```
./ossutil64 du oss://examplebucket --all-versions
```

The following output results are returned. The results show that 13 objects of 132,116,024 bytes in size are stored in examplebucket, in which 12 objects are Standard objects and one object is an Archive object.

```
storage class   object count            sum size(byte)
------------------------------------------------------------
Standard        12                      132115210
Archive         1                       814
------------------------------------------------------------
total object count: 13                          total object sum size: 132116024
total part count:   0                           total part sum size:   0
total du size(byte):132116024
0.382978(s) elapsed
```

Query the total size of the current versions of objects in the specified bucket

You can run the following command to query the total size of the current versions of objects in the dir directory in a bucket named examplebucket. The obtained size is measured in GB.

```
./ossutil64 du oss://examplebucket/dir/  --block-size GB
```

The following output results are returned. The results show that five Standard objects of 0.0002 GB in size are stored in the dir directory in examplebucket.

```
storage class   object count            sum size(byte)
------------------------------------------------------------
Standard        5                       232277
------------------------------------------------------------
total object count: 5                           total object sum size: 232277
total part count:   0                           total part sum size:   0
total du size(GB):0.0002
0.078757(s) elapsed
```

Query the total size of all version of objects whose names contain the specified prefix

You can run the following command to query the total size of all versions of objects whose names contain the "test" prefix in a bucket named examplebucket. The obtained size is measured in KB.

```
./ossutil64 du oss://examplebucket/test --all-versions --block-size KB
```

The following output results are returned. The results show that four Standard objects whose names contain the "test" prefix are stored in examplebucket and the total size of these objects is 448.1455 KB.

```
storage class   object count            sum size(byte)
-------------------------------------------------------------
Standard        4                       439425
-------------------------------------------------------------
total object count: 4                           total object sum size: 439425
total part count:   0                           total part sum size:   0
total du size(KB):448.1455
0.126340(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by multiple Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to query the total size of all versions of objects in a bucket named testbucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 du oss://testbucket --all-versions -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2N
bDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options, see Common options.

2.3.14. getallpartsize

This topic describes how to run the **getallpartsize** command to obtain the size of each part generated and the total size of all parts in incomplete multipart upload tasks.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about multipart upload, see Multipart upload and resumable upload.

Command syntax

```
./ossutil64 getallpartsize oss://bucketname
```

bucketname: specifies the name of the bucket that contains parts generated in incomplete multipart upload tasks.

Examples

The following command provides an example on how to list all parts that are generated in incomplete multipart upload tasks in the destination bucket named examplebucket:

```
./ossutil64 getallpartsize oss://examplebucket
```

If a similar output is displayed, information such as the number of parts, the total size of all parts, and the upload ID of the multipart upload task is obtained:

```
PartNumber      UploadId                                Size(Byte)      Path
1               F18A92392DFD4B3FA897C267829FE417        52428800        oss://examplebucket
/exampleobject.txt
2               F18A92392DFD4B3FA897C267829FE417        52428800        oss://examplebucket
/exampleobject.txt
3               F18A92392DFD4B3FA897C267829FE417        52428800        oss://examplebucket
/exampleobject.txt
4               F18A92392DFD4B3FA897C267829FE417        52428800        oss://examplebucket
/exampleobject.txt
5               F18A92392DFD4B3FA897C267829FE417        52428800        oss://examplebucket
/exampleobject.txt
6               F18A92392DFD4B3FA897C267829FE417        52428800        oss://examplebucket
/exampleobject.txt
total part count:6     total part size(MB):300.00
0.142115(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, if you want to query parts generated in incomplete multipart upload tasks in the testbucket bucket owned by another Alibaba Cloud account in the China (Hangzhou) region, run the following command:

```
./ossutil64 getallpartsize oss://testbucket -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDU
CV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options of this command, see Common options.

2.3.15. hash

This topic describes how to run the **hash** command to calculate the MD5 hash or CRC-64 of a local file.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 hash localfile [--type=<value>]
```

The following table describes the parameters that you can configure when you run the hash command.

Parameter	Description
localfile	The full path of the local file you want to upload.
--type	The method used to calculate the value of the local file. Default value: crc64. Valid values: • *md5*: If --type is set to md5, the MD5 hash and Content-MD5 value are displayed at the same time. The Content-MD5 value is obtained by calculating the MD5 hash to obtain a 128-bit number, and then encode the number in Base64. For more information about Content-MD5, see RFC 1864. • *crc64* The CRC-64 is calculated by using Standard ECMA-182.

After you upload the local file to the specified bucket, take note of the following items when you want to query or use the CRC-64 or MD5 hash:

- You can use the `X-Oss-Hash-Crc64ecma` field to query the CRC-64 or the `Content-Md5` field in the **stat** command to query the MD5 hash of the local file. For more information, see stat.
- If you upload a local file before Object Storage Service (OSS) supports CRC-64, the **stat** command cannot be run to query the CRC-64 of the local file.
- If you upload a local file by using append upload or multipart upload, the **stat** command cannot be run to query the Content-MD5 value of the local file.

Examples

- Calculate the CRC-64 of the `test.txt` local file

```
./ossutil64 hash test.txt --type=crc64
CRC64-ECMA                  : 295992936743767023
```

- Calculate the MD5 hash of the `test.txt` local file

```
./ossutil64 hash test.txt --type=md5
 MD5                        : 01C3C45C03B2AF225EFAD9F911A33D73
 Content-MD5                : AcPEXAOyryJe+tn5EaM9cw==
```

2.3.16. help

This topic describes how to run **help** to obtain the help information about commands. To obtain information about a command, we recommend that you run the **help** command.

Notice Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 help [command]
```

Examples

- Obtain the help information about all commands

  ```
  ./ossutil64 help
  ```

- Obtain the help information about the **cp** command

  ```
  ./ossutil64 help cp
  ```

- View the information about all options

  ```
  ./ossutil64 help -h
  ```

2.3.17. inventory (configure bucket inventories)

You can use the bucket inventory feature to export the information about specific objects in a bucket, such as the number, sizes, storage classes, and encryption status of the objects. Compared with the GetBucket (ListObjects) operation, we recommend that you use the bucket inventory feature to list a large number of objects. This topic describes how to run the **inventory** command to add, query, list, or delete bucket inventories.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about the bucket inventory feature, see Bucket inventory.

Add inventories

You can perform the following steps to add an inventory for a bucket:

1. Create a RAM role and authorize the role to read all objects in the source bucket and write objects in the destination bucket in which inventory lists are generated. For more information about how to create a RAM role, see Create a RAM role for a trusted Alibaba Cloud service.
2. Create a local file and configure inventories in the XML format in the file.
3. Use ossutil to read the inventories from the local file, and then add the inventories to the specified bucket.

- Command syntax

  ```
  ./ossutil64 inventory --method put oss://bucketname local_xml_file
  ```

  The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket for which you want to add inventories.

Parameter	Description
local_xml_file	The name of the local file in which the inventory is configured. Example: `local file.txt`.

- Examples

  i. Create a file named localfile.txt on the local computer and write different inventories to the file.

  The following example shows an inventory named inventorytest. Based on this inventory, OSS exports the information about all objects in a bucket named destbucket, including the storage classes, last update date, and multipart upload status of the objects. Exported inventory lists are encrypted by using the AES256 algorithm.

```
<?xml version="1.0" encoding="UTF-8"?>
  <InventoryConfiguration>
      <Id>inventorytest</Id>
      <IsEnabled>true</IsEnabled>
      <Filter></Filter>
      <Destination>
          <OSSBucketDestination>
              <Format>CSV</Format>
              <AccountId>1746495857602745</AccountId>
              <RoleArn>acs:ram::174649585760****:role/AliyunOSSRole</RoleArn>
              <Bucket>acs:oss:::destbucket</Bucket>
              <Encryption>
                  <SSE-OSS></SSE-OSS>
              </Encryption>
          </OSSBucketDestination>
      </Destination>
      <Schedule>
          <Frequency>Weekly</Frequency>
      </Schedule>
      <IncludedObjectVersions>All</IncludedObjectVersions>
      <OptionalFields>
          <Field>LastModifiedDate</Field>
          <Field>StorageClass</Field>
          <Field>IsMultipartUploaded</Field>
          <Field>ETag</Field>
          <Field>EncryptionStatus</Field>
          <Field>Size</Field>
      </OptionalFields>
  </InventoryConfiguration>
```

> **Notice** You can add multiple inventories for a bucket. The inventories are uniquely identified by their IDs. If the ID of the inventory you want to add is the same as that of an existing inventory, HTTP status code 409 is returned.

  ii. Add the inventories for a bucket named examplebucket.

```
./ossutil64 inventory --method put oss://examplebucket localfile.txt
```

  If a similar output is displayed, the inventories are added for examplebucket.

```
0.299514(s) elapsed
```

Query a specified inventory

- Command syntax

```
./ossutil64 inventory --method get oss://bucketname inventory_id [--local_xml_file ]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket whose inventories you want to query.
inventory_id	The name of the inventory you want to query.
local_xml_file	The name of the local file used to store the inventory obtained inventory. Example: `localfile.txt`. If this parameter is not specified, obtained inventories are displayed without being stored in a local file.

- Examples

```
./ossutil64 inventory --method get oss://examplebucket inventorytest localfile.txt
```

If a similar output is displayed, an inventory named inventorytest is configured for the bucket named examplebucket, and the inventory is written to a local file named localfile.txt.

```
0.212407(s) elapsed
```

Query all inventory rules configured for a bucket

- Command syntax

```
./ossutil64 inventory --method list oss://bucketname [--local_xml_file ] [--marker <value
>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket of which inventories you want to query.
local_xml_file	The name of the local XML file used to store the obtained inventories. If this parameter is not specified, obtained inventories are displayed without being stored in a local file.
marker	The filtering conditions for inventories. Inventory lists are generated for only objects whose names contain the specified prefix. If this parameter is not specified, inventory lists are generated for all objects in the bucket.

- Examples

```
./ossutil64 inventory --method list oss://examplebucket localfile.txt dest
```

If a similar output is displayed, all inventories that are configured for a bucket named examplebucket and apply to objects whose names contained the dest prefix are obtained and written to a local file named localfile.txt.

```
0.216897(s) elapsed
```

Delete a specified inventory

- Command syntax

```
./ossutil64 inventory --method delete oss://bucketname inventory_id
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket for which you want to delete the inventory.
inventory_id	The name of the inventory you want to delete.

- Examples

```
./ossutil64 inventory --method delete oss://examplebucket inventorytest
```

If a similar output is displayed, an inventory named inventorytest is deleted for a bucket named examplebucket.

```
0.212407(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to configure inventories for a bucket named examplebucket, which is located in the China (Hangzhou) region and owned by another Alibaba cloud account:

```
./ossutil64 inventory --method put oss://examplebucket local_xml_file -e oss-cn-hangzhou.al
iyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that apply to the inventory command, see Common options.

2.3.18. lifecycle

After you configure lifecycle rules for a bucket, OSS (Object Storage Service) regularly converts the storage class of objects in the bucket to Infrequent Access (IA), Archive, or Cold Archive, or deletes expired objects and parts to save your storage costs. This topic describes how to run the **lifecycle** command to add, modify, query, and delete lifecycle rules.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace the **./ossutil64** part in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about lifecycle rules, see Lifecycle rules based on the last modified time.

Add or modify lifecycle rules

You can perform the following steps to add or modify lifecycle rules:

1. Create a local file and configure lifecycle rules in the XML format in the file.
2. Use ossutil. ossutil reads the lifecycle rule configurations from the local file, and then adds the configurations to the specified bucket.

- Command syntax

```
./ossutil64 lifecycle --method put oss://bucketname local_xml_file
```

The following table describes the parameters that you can configure when you run this command to add or modify lifecycle rules.

Parameter	Description
bucketname	The name of the bucket for which you want to add or modify lifecycle rules.
local_xml_file	The name of the local file in which the lifecycle rule is configured. Example: `localfile.txt`.

- Examples

> **Notice** You can add multiple lifecycle rules for a bucket. Each rule is identified by their unique IDs. If the ID of the lifecycle rule you want to add already exists, the HTTP status code 409 is returned.

i. Create a file named `localfile.txt` on your local computer and configure lifecycle rules based on your requirements.

The following examples show how to configure some common lifecycle rules:

- Example 1

  Specify that objects in the bucket named examplebucket are deleted 365 days after they are last modified, and the storage class of objects whose names contain the test/ prefix is converted to Archive 30 days after they are last modified.

  For more information about how to convert the storage class of objects by using lifecycle rules, see Configure lifecycle rules to automatically convert the storage class of an object.

```
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule1</ID>
    <Prefix></Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>test-rule2</ID>
    <Prefix>test/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>Archive</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

- Example 2

  Specify that objects that are last modified before December 30, 2019 in the bucket named examplebucket expire.

```
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule0</ID>
    <Prefix></Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <CreatedBeforeDate>2019-12-30T00:00:00.000Z</CreatedBeforeDate>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

- Example 3

  Specify that the storage class of objects in the versioning-enabled bucket named examplebucket is converted to IA 10 days after they are last modified. In addition, the storage class of objects is converted to Archive 60 days after they become a previous version, and objects are deleted 90 days after they become a previous version.

```
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>test-rule3</ID>
    <Prefix></Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>10</Days>
      <StorageClass>IA</StorageClass>
    </Transition>
    <NoncurrentVersionTransition>
      <NoncurrentDays>60</NoncurrentDays>
      <StorageClass>Archive</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>90</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

ii. Add lifecycle rules for the bucket named examplebucket.

```
./ossutil64 lifecycle --method put oss://examplebucket localfile.txt
```

If a similar output is displayed, the lifecycle rules are added to the bucket:

```
0.299514(s) elapsed
```

Query lifecycle rules

- Command syntax

```
./ossutil64 lifecycle --method get oss://bucketname [local_xml_file]
```

The following table describes the parameters that you can configure when you run this command to query lifecycle rules configured for a bucket.

Parameter	Description
bucketname	The name of the bucket whose lifecycle rules you want to query.
local_xml_file	The name of the local file used to store obtained lifecycle rule configurations. Example: `localfile.txt`. If this parameter is not specified, the obtained lifecycle rule configurations are displayed on the screen.

- Examples

You can run the following command to query the lifecycle rules configured for the bucket named examplebucket:

```
./ossutil64 lifecycle --method get oss://examplebucket localfile.txt
```

If a similar output is displayed, the lifecycle rules configured for the bucket named examplebucket are obtained and written to the localfile.txt file:

```
0.212407(s) elapsed
```

Delete lifecycle rules

- Command syntax

```
./ossutil64 lifecycle --method delete oss://bucketname
```

- Examples

  You can run the following command to delete the lifecycle rules configured for the bucket named examplebucket:

```
./ossutil64 lifecycle --method delete oss://examplebucket
```

  If a similar output is displayed, the lifecycle rules configured for the bucket named examplebucket are deleted:

```
0.530750(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of a specific bucket. To use command-line tool ossutil to manage buckets within multiple Alibaba Cloud accounts, you can use the -i option to switch to the AccessKey ID of a specific account and use the -k option to switch to the AccessKey secret of the account.

For example, you can run the following command to configure lifecycle rules for a bucket named examplebucket, which is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 lifecycle --method put oss://examplebucket localfile.txt -e oss-cn-hangzhou.ali
yuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that apply to the lifecycle command, see Common options.

2.3.19. listpart

This topic describes how to run the **listpart** command to list parts in an object that are generated in a multipart upload task that is not completed.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about multipart upload, see Multipart upload.

Command syntax

```
./ossutil64 listpart oss://bucketname/objectname uploadid
```

The following table describes the parameters that you can configure when you run the listpart command.

Parameter	Description
bucketname	The name of the bucket to which you want to upload the object by using multipart upload.
objectname	The name of the object for which the multipart upload task is not completed.
uploadid	The Object Storage Service (OSS) server returns a globally unique upload ID for the InitiateMultipartUpload operation. You can use the upload ID to uniquely identify the multipart upload task. You can also use an upload ID to cancel or query a multipart upload task.

> **Notice** To obtain the name of the object for which the multipart upload task is not completed, the time when the multipart upload task was initiated, and the upload ID of the multipart upload task, you can specify the `-m` option in the **ls** command. For more information, see ls.

Examples

You can run the following command to list the parts in the exampleobject.txt object of the examplebucket bucket that are generated when the multipart upload task is not completed:

```
./ossutil64 listpart oss://examplebucket/exampleobject.txt 89A46B10E94A4ED5A7E9869F9409****
```

If a similar output is displayed, information such as the number of parts in the exampleobject.txt object, ETag values, and part size is obtained.

```
PartNumber      Etag                                Size(Byte)      LastModifyTime
2               "BD106FED29B349A635BE289746DB****"  4443891         2021-07-21 09:01:33
3               "25DA5C7BB933DBD732B6C11111EB****"  4443891         2021-07-21 09:01:31
4               "AFDA91A8D4F476BAC972306873A3****"  4443891         2021-07-21 09:01:31
11              "2FFEE8D5049AB059D7EC801927BB****"  4443891         2021-07-21 09:01:32
total part count:4      total part size(MB):16.95
0.101978(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to query the information about the parts in testobject.txt that are generated when the multipart upload task is not completed. The testobject.txt object is in the testbucket bucket that is located in the China (Hangzhou) region and owned by another Alibaba Cloud account.

```
./ossutil64 listpart oss://testbucket/testobject.txt 46C56B10E94A4ED6G8U9869F9409**** -e os
s-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the listpart command, see Common options.

2.3.20. logging

When you access Object Storage Service (OSS), large numbers of access logs are generated. After you enable and configure logging for a bucket, OSS generates log objects every hour in accordance with predefined naming conventions and then stores the access logs as objects in a specified bucket. You can run the **logging** command to enable or disable logging for a bucket, and query the logging configurations of a bucket.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about logging, see Log storage.

Enable logging for a bucket

If logging is not enabled for a bucket, you can run this command to enable logging for this bucket. If logging is enabled for this bucket, the existing logging configurations of this bucket are overwritten when you run this command.

Enable or modify logging configurations for a bucket by using the following command syntax:

- Command syntax

  ```
  ./ossutil64 logging --method put oss://bucketname   oss://targetbucket/[prefix]
  ```

  The following table describes the parameters that you can configure when you run this command to enable logging for a bucket.

Parameter	Description
bucketname	The name of the source bucket for which log objects are generated.
targetbucket	The name of the destination bucket in which log objects are stored.

Parameter	Description
prefix	The directory in which log objects are stored. If you specify this parameter, log objects are stored in a specified directory of the destination bucket. If you do not specify this parameter, log objects are stored in the root directory of the destination bucket.

> **Notice** The source bucket for which log objects are generated and the destination bucket where the generated log objects are stored can be the same bucket or different buckets. However, the destination bucket must belong to the same account in the same region as the source bucket.

- Examples

  Store log objects generated for the srcbucket source bucket in the root directory of the destbucket destination bucket.

```
./ossutil64 logging --method put oss://srcbucket  oss://destbucket
```

  Store log objects generated for the srcbucket source bucket in the destdir subdirectory of the root directory of the destbucket destination bucket.

```
./ossutil64 logging --method put oss://srcbucket  oss://destbucket/destdir
```

  If a similar output is displayed, logging is enabled for the srcbucket bucket:

```
0.098601(s) elapsed
```

View the logging configurations of a bucket

- Command syntax

```
./ossutil64 logging --method get oss://bucketname [local_xml_file]
```

  The following table describes the parameters that you can configure when you run this command to view the logging configurations of a bucket.

Parameter	Description
bucketname	The name of the bucket for which you want to query the logging configurations.
local_xml_file	The name of the local file used to store the logging configurations. Example: `localfile.txt`. If this parameter is not specified, obtained logging configurations are displayed without being stored in a local file.

- Examples

- Query the logging configurations of the examplebucket bucket and write the obtained configuration results to the `localfile.txt` local file.

  ```
  ./ossutil64 logging --method get oss://examplebucket localfile.txt
  ```

  If a similar output is displayed, you have obtained the logging configurations:

  ```
  0.212407(s) elapsed
  ```

- Query the logging configurations of the examplebucket bucket and display the results without storing these results in a local file.

  ```
  ./ossutil64 logging --method get oss://examplebucket
  ```

  If a similar output is displayed, the log objects of the examplebucket bucket are stored in the root directory of the destbucket bucket:

  ```
  <?xml version="1.0" encoding="UTF-8"?>
    <BucketLoggingStatus>
        <LoggingEnabled>
            <TargetBucket>destbucket</TargetBucket>
            <TargetPrefix></TargetPrefix>
        </LoggingEnabled>
    </BucketLoggingStatus>
  0.109979(s) elapsed
  ```

Disable logging for a bucket

Before you disable logging, OSS keeps generating log objects. Delete log objects you no longer need to reduce storage costs based on lifecycle rules. For more information, see Lifecycle rules based on the last modified time.

- Command syntax

  ```
  ./ossutil64 logging --method delete oss://bucketname
  ```

  bucketname indicates the name of the bucket for which you want to disable logging.

- Examples

  Disable logging for the examplebucket bucket.

  ```
  ./ossutil64 logging --method delete oss://examplebucket
  ```

  If a similar output is displayed, logging is disabled for the examplebucket bucket:

  ```
  0.212409(s) elapsed
  ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

The following command provides an example on how to transfer log objects generated for the testbucket source bucket in the China (Hangzhou) region to the destbucket destination bucket:

```
./ossutil64 logging --method put oss://testbucket  oss://destbucket -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the logging command, see Common options.

2.3.21. lrb

You can run the **lrb** command to obtain the basic information about buckets within a region or across multiple regions. The basic bucket information includes bucket names, creation time, storage classes, and the number of buckets.

Notice Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 lrb conf_file [-e <value>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
conf_file	If you want to obtain the information of buckets across multiple regions, specify a local configuration file and enter the endpoints of these regions in the configuration file. Separate multiple endpoints with line feeds.
-e	If you want to obtain the information of buckets within a region, set this parameter to the endpoint of the region.

Examples

- Obtain the information of buckets within a region.

  Obtain the information of buckets within the China (Hangzhou) region.

  ```
  ./ossutil64 lrb -e oss-cn-hangzhou.aliyuncs.com
  ```

  Obtain the information of buckets within the region whose endpoint is specified in the ossutil configuration file. For more information about the endpoint specified in the ossutil configuration file, see config.

  ```
  ./ossutil64 lrb
  ```

- Obtain the information of buckets across multiple regions.

  i. Create a local file and specify the endpoints of these regions in the local file.

Create a local file named `localfile.txt` and enter the endpoints of these regions in which the buckets are located based on the scenario. The following code provides an example on how to configure `localfile.txt`:

```
oss-cn-hangzhou-aliyuncs.com
oss-cn-shenzhen-aliyuncs.com
oss-cn-shanghai-aliyuncs.com
```

ii. Obtain the information of buckets in the China (Hangzhou), China (Shenzhen) and China (Shanghai) regions.

```
./ossutil64 lrb localfile.txt
```

- Output

  After you obtain the information of buckets, the output includes the bucket names, creation time, storage classes, number of buckets, and the time Object Storage Service (OSS) took to return this output. The following output indicates the information of buckets across multiple regions:

```
CreationTime                              Region    StorageClass    BucketName
2021-07-06 14:21:09 +0800 CST       oss-cn-hangzhou        Standard    oss://examplebucke
t1
2021-07-06 14:21:44 +0800 CST       oss-cn-hangzhou        Standard    oss://examplebucke
t2
2021-06-16 18:32:32 +0800 CST       oss-cn-shanghai        Standard    oss://examplebucke
t3
2021-06-30 16:04:41 +0800 CST       oss-cn-shanghai        Standard    oss://examplebucke
t4
2021-07-07 12:33:35 +0800 CST       oss-cn-shenzhen        Standard    oss://examplebucke
t5
Bucket Number is: 5
0.124193(s) elapsed
```

Common options

To use ossutil to manage buckets in different regions, you can use the -e option to switch to the endpoint to which the specified bucket belongs. To use ossutil to manage buckets that belong to multiple Alibaba Cloud accounts, you can switch to the AccessKey ID of a specified account by using the -i option and switch to the AccessKey secret of the specified account by using the -k option.

The following sample command shows how to obtain the information of buckets in the China (Shanghai) region of another Alibaba Cloud account:

```
./ossutil64 lrb -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7Ea
mOjy2W5RVAHUY9H****
```

For more information about other common options of this command, see Common options.

2.3.22. ls

You can run the ls command to list buckets, objects, and parts of the current Alibaba Cloud account.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

List buckets

- Command syntax

```
./ossutil64 ls [-s] [--limited-num] [--marker]
```

The following table describes the options that you can configure in the command to list buckets.

Option	Description
-s	Specifies that only bucket names are listed.
--limited-num	Specifies the maximum number of buckets to list. You can specify this parameter together with the marker parameter to display returned results by page.
--marker	Specifies the position from which the list operation starts. Buckets whose names are alphabetically after the value of marker are listed.

- Examples
  - List all buckets that belong to the current Alibaba Cloud account.

```
./ossutil64 ls
```

You can also run the following command to list all buckets owned by the current account:

```
./ossutil64 ls oss://
```

The following output indicates that all buckets that belong to the current Alibaba Cloud account are listed. Listed information includes the bucket names, creation time, regions, storage classes, and number of the buckets.

```
2016-10-21 16:18:37 +0800 CST       oss-cn-hangzhou         Archive    oss://examplebuc
ketA
2016-12-01 15:06:21 +0800 CST       oss-cn-hangzhou        Standard    oss://examplebuc
ketB
2016-07-20 10:36:24 +0800 CST       oss-cn-hangzhou              IA    oss://examplebuc
ketC
2016-10-21 17:31:27 +0800 CST       oss-cn-hangzhou         Archive    oss://examplebuc
ketD
Bucket Number is:4
0.252174(s) elapsed
```

- List all buckets in simple mode.

  ```
  ./ossutil64 ls -s
  ```

  The following output indicates that all buckets that belong to the current Alibaba Cloud account are listed. Listed information includes only the bucket names and the number of the buckets.

  ```
  oss://examplebucketA
  oss://examplebucketB
  oss://examplebucketC
  oss://examplebucketD
  Bucket Number is:4
  0.235104(s) elapsed
  ```

- List buckets whose names are alphabetically after examplebucketA, which is specified by the marker parameter.

  ```
  ./ossutil64 ls oss:// --limited-num=2 -s --marker examplebucketA
  ```

  The following output indicates that two buckets whose names are alphabetically after examplebucketA are listed:

  ```
  2016-12-01 15:06:21 +0800 CST       oss-cn-hangzhou        Standard    oss://examplebuc
  ketB
  2016-07-20 10:36:24 +0800 CST       oss-cn-hangzhou              IA    oss://examplebuc
  ketC
  Bucket Number is:2
  0.132174(s) elapsed
  ```

List objects

- Command syntax

  ```
  ./ossutil64 ls oss://bucketname[/prefix] [-s] [-d] [--limited-num] [--marker] [--include]
  [--exclude]  [--version-id-marker] [--all-versions]
  ```

  The following table describes the options that you can configure in the command to list objects.

Option	Description
bucketname	Specifies the name of the bucket in which objects you want to list are stored.
prefix	Specifies the prefix of objects that you want to list. Specifies the parameter when you want to list objects whose names contain the specified prefix.
-s	Specifies that only object names are listed.
-d	Specifies that only objects and subdirectories in the root directory of the bucket are listed. Objects in subdirectories are not listed.
--limited-num	Specifies the maximum number of objects to list. You can specify this parameter together with the marker parameter to display returned results by page.

Option	Description
--marker	Specifies the position from which the list operation starts. Objects whose names are alphabetically after the value of marker are listed.
--include	Specifies that objects that meet specified conditions are listed. For example, a value of `*.jpg` indicates that all objects in the JPG format are listed. For more information, see Batch upload objects that meet specified conditions.
--exclude	Specifies that objects that do not meet specified conditions are listed. For example, a value of `*.txt` indicates that all objects that are not in the TXT format are listed.
--version-id-marker	Specifies the position from which the list operation starts. Object versions whose IDs are alphabetically after the value of marker are listed. You can specify this parameter only when versioning is enabled for the bucket.
--all-versions	Specifies that all versions of objects in the bucket are listed. You can specify this parameter only when versioning is enabled for the bucket.

- Examples
  - List all objects in a bucket named examplebucket.

```
./ossutil64 ls oss://examplebucket
```

The following output indicates that all objects in the examplebucket bucket are listed. Listed information includes the last modified time, sizes, ETag values, and names of the objects.

The ETag value of an object is used to identify the content of the object. If an object is created by using a PutObject request, the ETag of the object is the MD5 hash of the object content. If an object is created by using other methods, the ETag of the object is not the MD5 hash of the object content but a unique value calculated based on the object.

```
LastModifiedTime                   Size(B)  StorageClass   ETag
ObjectName
2020-12-01 15:06:37 +0800 CST          114      Standard   61DE142E5AFF9A6748707D4A77B
FBCFB        oss://examplebucket/example.txt
2020-12-01 15:06:42 +0800 CST       363812      Standard   E7581E5D2EBC56ECCB6FB6050B4
C6545        oss://examplebucket/examplefolder/photo.jpg
2020-12-01 15:06:45 +0800 CST     57374182      Standard   BE97B7AD7A2C1277B11221E5C95
37544        oss://examplebucket/video.mp4
Object Number is:3
0.007379(s) elapsed
```

- List objects whose names contain the "example" prefix in a bucket named examplebucket.

```
./ossutil64 ls oss://examplebucket/example
```

The following output indicates that all objects whose names contain the "example" prefix in the examplebucket bucket are listed:

```
LastModifiedTime                   Size(B)  StorageClass   ETag
ObjectName
2020-12-01 15:06:37 +0800 CST          114      Standard   61DE142E5AFF9A6748707D4A77B
FBCFB        oss://examplebucket/example.txt
2020-12-01 15:06:42 +0800 CST       363812      Standard   E7581E5D2EBC56ECCB6FB6050B4
C6545        oss://examplebucket/examplefolder/photo.jpg
Object Number is:2
0.007379(s) elapsed
```

- List objects whose names contain the ".mp4" suffix in a bucket named examplebucket.

```
./ossutil64 ls oss://examplebucket --include *.mp4
```

The following output indicates that objects whose names contain the ".mp4" suffix in the examplebucket bucket are listed:

```
LastModifiedTime                    Size(B)  StorageClass   ETag
ObjectName
2020-12-01 15:06:45 +0800 CST      57374182      Standard   BE97B7AD7A2C1277B11221E5C95
37544        oss://examplebucket/video.mp4
Object Number is:1
0.007379(s) elapsed
```

- List only objects and subdirectories in the root directory of a bucket named examplebucket.

```
./ossutil64 ls oss://examplebucket -d
```

The following output indicates that objects and subdirectories in the root directory of the examplebucket bucket are listed:

```
oss://examplebucket/example.txt
oss://examplebucket/examplefolder/
oss://examplebucket/video.mp4
Object and Directory Number is: 3
0.278489(s) elapsed
```

- List all versions of all objects in a bucket named examplebucket.

```
./ossutil64 ls oss://examplebucket --all-versions
```

The following output indicates that all versions of all objects in the examplebucket bucket are listed:

```
LastModifiedTime                   Size(B)  StorageClass   ETag
VERSIONID                                                          IS-LATEST   DELETE-
MARKER   ObjectName
2020-12-01 15:06:37 +0800 CST         114      Standard   61DE142E5AFF9A6748707D4A77BFB
CFB      CAEQARiBgICUsOuR2hYiIDI3NWVjNmEyYmM0NTRkZWNiMTkxY2VjMDMwZjFlMDA3    true
false           oss://examplebucket/example.txt
2020-06-11 11:03:37 +0800 CST      363812      Standard   E7581E5D2EBC56ECCB6FB6050B4C6
545      CAEQARiBgIDZtvuR2hYiIDNhYjRkN2M5NTA5OTRlN2Q4YTYzODQwMzQ4NDYwZDdm    true
false           oss://examplebucket/examplefolder/photo.jpg
2021-01-26 13:27:08 +0800 CST           0
CAEQLxiBgIDd7NH0uRciIDA3Yzg0MTZjOWNlYzQ4ODZhMzVkZWE0MmE2NzBlYTYx    true        true
oss://examplebucket/image.png
2020-12-01 15:06:45 +0800 CST    57374182      Standard   BE97B7AD7A2C1277B11221E5C9537
544      CAEQLBiBgMDZiprwthciIDY2NGM0NTNmZDE3ODRmZmVhZGM4YTUwZGQyNGU3ZjQ3    true
false           oss://examplebucket/video.mp4
2016-06-11 10:53:46 +0800 CST      118076      Standard   FFDB300F053AAF06F4C4C58A4869C
427      CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3MDRk    false
false           oss://examplebucket/example.txt
2016-06-11 11:02:05 +0800 CST      345374      Standard   078A9852BCF81DC4811E6EDCBFD12
1BE      CAEQARiBgICNz_iR2hYiIGJjZTBjNDQxYWRhNTQ2ZTNiNmMzYzQ1YzMzMDA5ZjUw    false
false           oss://examplebucket/examplefolder/photo.jpg
Object Number is: 6
0.692000(s) elapsed
```

- List all versions of an object named *example.txt* in the root directory of a bucket named examplebucket.

```
./ossutil64 ls oss://examplebucket/example.txt --all-versions
```

The following output indicates that all versions of the *example.txt* object are listed:

```
LastModifiedTime                   Size(B)  StorageClass  ETag
VERSIONID                                                          IS-LATEST   DELETE-
MARKER  ObjectName
2020-12-01 15:06:37 +0800 CST         114      Standard   61DE142E5AFF9A6748707D4A77BFB
CFB      CAEQARiBgICUsOuR2hYiIDI3NWVjNmEyYmM0NTRkZWNiMTkxY2VjMDMwZjFlMDA3    true
false           oss://examplebucket/example.txt
2016-06-11 10:53:46 +0800 CST         114      Standard   61DE142E5AFF9A6748707D4A77BFB
CFB      CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3MDRk    false
false           oss://examplebucket/example.txt
Object Number is: 2
0.361000(s) elapsed
```

List parts

- Command syntax

```
./ossutil64 ls oss://bucketname[/prefix] [-s] [-d] [-m] [-a] [--limited-num] [--upload-id-marker]
```

The following table describes the options you can configure in the command to list parts.

Option	Description
bucketname	Specifies the name of the bucket in which parts you want to list are stored. Add this option when you want to list objects in a specified bucket.
prefix	Specifies the parts whose names contain a specified prefix are listed.
-s	Specifies that only upload IDs and objects names are listed.
-d	Specifies that only objects and subdirectories in the root directory of the bucket are listed. Objects in subdirectories are not listed.
-m	Specifies that the command is run to list parts.
-a	Specifies that the command is run to list objects and parts.
--limited-num	Specifies the maximum number of parts to list. You can specify this parameter together with the --upload-id-marker parameter to display returned results by page.
--upload-id-marker	Specifies the position from which the list operation starts. Parts whose upload IDs are alphabetically greater than the value of marker are listed.

- Examples
  - List all parts in a bucket named examplebucket.

    ```
    ./ossutil64 ls oss://examplebucket -m
    ```

    The following output indicates that all parts in the examplebucket bucket are listed:

    ```
    InitiatedTime                    UploadID                            ObjectName
    2017-01-13 03:45:26 +0000 CST    15754AF7980C4DFB8193F190837520BB    oss://examplebucket/test.mp4
    2017-01-13 03:45:13 +0000 CST    2A1F9B4A95E341BD9285CC42BB950EE0    oss://examplebucket/test.mp4
    2017-01-13 03:45:01 +0000 CST    3998971ACAF94AD9AC48EAC1988BE863    oss://examplebucket/test.mp4
    2017-01-20 11:16:21 +0800 CST    A20157A7B2FEC4670626DAE0F4C0073C    oss://examplebucket/object.exe
    UploadId Number is:4
    0.191289(s) elapsed
    ```

- List all objects and parts in a bucket named examplebucket.

```
./ossutil64 ls oss://examplebucket -a
```

The following output indicates that all objects and parts in the examplebucket bucket are listed:

```
LastModifiedTime                   Size(B)  StorageClass   ETag
ObjectName
2020-12-01 15:06:37 +0800 CST          114      Standard   61DE142E5AFF9A6748707D4A77B
FBCFB        oss://examplebucket/example.txt
2020-12-01 15:06:42 +0800 CST       363812      Standard   E7581E5D2EBC56ECCB6FB6050B4
C6545        oss://examplebucket/examplefolder/photo.jpg
2020-12-01 15:06:45 +0800 CST     57374182      Standard   BE97B7AD7A2C1277B11221E5C95
37544        oss://examplebucket/video.mp4
Object Number is:3
InitiatedTime                     UploadID                           ObjectName
2017-01-13 03:45:26 +0000 CST     15754AF7980C4DFB8193F190837520BB    oss://examplebuck
et/test.mp4
2017-01-13 03:45:13 +0000 CST     2A1F9B4A95E341BD9285CC42BB950EE0    oss://examplebuck
et/test.mp4
2017-01-13 03:45:01 +0000 CST     3998971ACAF94AD9AC48EAC1988BE863    oss://examplebuck
et/test.mp4
2017-01-20 11:16:21 +0800 CST     A20157A7B2FEC4670626DAE0F4C0073C    oss://examplebuck
et/object.exe
UploadId Number is:4
0.791289(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of a specified bucket. To use ossutil to manage buckets that belong to multiple Alibaba Cloud accounts, you can switch to the AccessKey ID of a specified account by using the -i option and switch to the AccessKey secret of the specified account by using the -k option.

For example, you can run the following command to list all objects in a bucket named test, which is located in the China (Hangzhou) region and owned by another Alibaba Cloud account:

```
./ossutil64 ls oss://test -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 6
7DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the ls command, see Common options.

2.3.23. mb (create buckets)

A bucket is a container for objects stored in OSS. You must create a bucket before you upload an object to OSS. This topic describes how to run the **mb** command to create buckets.

Notice Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 mb oss://bucketname [--acl <value>][--storage-class <value>][--redundancy-type <value>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket that you want to create. A bucket name must be globally unique within Object Storage Service (OSS). The name of a bucket cannot be changed after the bucket is created.
--acl	The access control list (ACL) of the bucket. Default value: private. Valid values: • *private*: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. • *public-read*: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. If you set the bucket ACL to this value, data leakage may occur and you may be charged additional fees. Exercise caution when you set the bucket ACL to this value. • *public-read-write*: All users, including anonymous users, can perform read and write operations on the objects in the bucket. If you set the bucket ACL to this value, data leakage may occur and you may be charged additional fees. If a user writes illegal information to your objects, your legitimate interests and rights may be infringed. We recommend that you do not set the bucket ACL to this value except in special scenarios.
--storage-class	The storage class of the bucket. Default value: Standard. Valid values: • *Standard*: This storage class can handle frequent data access. • *IA*: This storage class is suitable for long-term storage of data that is infrequently accessed (once or twice each month). Objects of the IA storage class have a minimum storage period of 30 days and a minimum billable size of 64 KB. You can access objects of the IA storage class in real time. You are charged for data retrieval fees when you access IA objects. • *Archive*: This storage class applies to scenarios that store data for a long period of time. Objects of the Archive storage class have a minimum storage period of 60 days and a minimum billable size of 64 KB. You must restore an Archive object before you can access it. The restoration takes about one minute, and you are charged for data retrieval fees. • *ColdArchive*: This storage class is suitable for long-term storage of data that is barely accessed. Objects of the Cold Archive storage class have a minimum storage period of 180 days and a minimum billable size of 64 KB. You must restore an object of the Cold Archive storage class before you can access it. The time required to restore a Cold Archive object depends on the object size and the restoration mode. You are charged for data retrieval fees when you restore Cold Archive objects. For more information about storage classes, see Overview.

Parameter	Description
--redundancy-type	The redundancy type of the bucket. Valid values: • *LRS*: If you set the redundancy type of a bucket to locally redundant storage (LRS), OSS stores the copies of each object across different devices within the same zone. This way, OSS ensures data reliability and availability when hardware failures occur. • *ZRS*: If you set the redundancy type of a bucket to zone-redundant storage (ZRS), OSS uses the multi-zone mechanism to distribute user data across three zones within the same region. This way, the data can be accessed even if one zone becomes unavailable due to failures such as power outages and fires. **Notice** ZRS is supported in the following regions: China (Shenzhen), China (Beijing), China (Hangzhou), China (Shanghai), China (Hong Kong), and Singapore.

Examples

- You can run the following command to create a bucket named examplebucket:

```
./ossutil64 mb oss://examplebucket
```

If you do not specify the region in which you want to create the bucket, the bucket is created in the region specified by the endpoint in the ossutil configuration file. For example, if the endpoint specified in the configuration file is `https://oss-cn-hangzhou.aliyuncs.com`, the bucket is created in the China (Hangzhou) region.

- You can run the following command to create a bucket named examplebucket and set the bucket ACL to private, storage class to IA, and redundancy type to ZRS.

```
./ossutil64 mb oss://examplebucket --acl private --storage-class IA --redundancy-type ZRS
```

- If a similar output is displayed, the specified bucket is created.

```
0.335189(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to create a bucket named examplebucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account.

```
./ossutil64 mb oss://examplebucket -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA*
***  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information the mb command, see Common options.

2.3.24. mkdir

To organize objects in a bucket, you can use directories. This topic describes how to run the **mkdir** command to create directories in ossutil.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 mkdir oss://bucketname /dirname [--encoding-type <value>]
```

The following table describes the parameters that you can configure when you run this command to create directories.

Parameter	Description
bucketname	The name of the bucket in which you want to create directories.
dirname	The name of the directory that you want to create. A directory name must end with a forward slash (/). If you do not end the directory name with a forward slash (/), ossutil automatically adds one at the end of the name.
--encoding-type	The encoding type used to encode the directory name, which is specified by the part that follows `oss://bucket_name` in the value of dirname. Valid value: *url*. If you do not specify this parameter, the directory name is not encoded.

Examples

You can perform the following steps to upload an object to a specified directory:

1. Create a directory.
   - Create a single-level directory

     ```
     ./ossutil64 mkdir oss://examplebucket/dir/
     ```

     If a similar output is returned, a directory named dir/ is created in the bucket named examplebucket:

     ```
     0.385877(s) elapsed
     ```

   - Create a multi-level directory

     To further classify objects stored in a directory, you can create multi-level directories. For example, you can run the following command to create a multi-level directory oss://examplebucket/Photo/2021/ to store the snapshots that are generated in 2021:

     ```
     ./ossutil64 mkdir oss://examplebucket/Photo/2021/
     ```

If you accidentally delete the 2021/ directory and no other objects are stored in the Photo/ directory, the Photo/ directory is also deleted.

2. Upload an object to a directory

   You can run the following command to upload an object named exampleobject.txt to the dir/ directory in a bucket named examplebucket:

   ```
   ./ossutil64 cp exampleobject.txt oss://examplebucket/dir/
   ```

   If a similar output is returned, the object is uploaded to the specified directory:

   ```
   Succeed: Total num: 1, size: 0. OK num: 1(upload 1 files).
   average speed 0(byte/s)
   ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. When you use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to create a directory named dir/ in a bucket named examplebucket, which is located in the China (Hangzhou) region and owned by another Alibaba Cloud account:

```
./ossutil64 mkdir oss://examplebucket/dir/ -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUC
V8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the mkdir command, see Common options.

2.3.25. object-tagging (add, modify, query, and delete object tags)

Object Storage Service (OSS) allows you to use tags to classify objects. You can batch manage objects that have the same tag. For example, you can specify the validity period of the objects or convert the storage class of the objects that have the same tag. The **object-tagging** command is used to add, modify, query, and delete object tags.

Usage notes

- Binary

  Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

- Tag synchronization

  When you perform cross-region replication (CRR), you can synchronize objects that have specified tags from the source bucket to the destination bucket. For more information, see Configure CRR.

For more information about object tagging, see Object tagging in OSS Developer Guide.

Command syntax

```
./ossutil64 object-tagging oss://bucketname[/prefix][key#value]
--method <value>
[--encoding-type <value>]
[-r, --recursive]
[--payer <value>]
[--version-id <value>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket in which the objects for which you want to configure tagging are stored.
prefix	The resources in the bucket, such as directories and objects.
key	The key of the tag you want to configure. The object tagging feature uses a key-value pair to tag an object. You can add up to 10 tags to each object. The tags of the same object must have unique tag keys. The key of a tag must comply with the following conventions: • The key of a tag is up to 128 characters in length and is case-sensitive. • The key of a tag can contain letters, digits, spaces, and the following special characters: +=._:/
value	The value of the tag you want to configure. The value of a tag must comply with the following conventions: • The value of a tag is up to 256 characters in length and is case-sensitive. • The value of a tag can contain letters, digits, spaces, and the following special characters: +=._:/
--method	The type of the request. Valid values: • *put*: The command to add tags to an object or modify the tags of an object. • *get*: The command to query the tags of an object. • *delete*: The command to delete the tags of an object.
--encoding-type	The method used to encode the prefix specified by the object name that follows `oss://bucket_name` in the full object path. Valid value: *url*. If this parameter is not specified, the prefix is not encoded.
-r, --recursive	If you specify this parameter in the command, ossutil configures tagging for all objects whose names contain the prefix specified by the prefix parameter. If you do not specify this parameter in the command, ossutil configures tagging only for the specified object.

Parameter	Description
--version-id	The ID of the version of the object for which you want to configure tagging. This parameter applies only to objects in buckets for which versioning is enabled or suspended.
--payer	The payer of the traffic and request fees incurred during queries. If you want that the requester who accesses the resources in the specified path to be charged for the traffic and request fees incurred when the command is run, set this parameter to *requester*.

Add or modify object tags

Only the owner of a bucket and RAM users that have the PutObjectTagging permission can add tags to objects in the bucket or modify the tags of objects in the bucket.

The following examples show how to add and modify tags of an object:

> **Notice** In the following examples, if specified object does not have tags, the tags are added. If the specified object already has tags, the existing tags are replaced.

- Configure a tag whose key is tagkey and whose value is tagvalue for an object named exampleobject.txt in a bucket named examplebucket.

  ```
  ./ossutil64 object-tagging --method put oss://examplebucket/exampleobject.txt tagkey#tagvalue
  ```

- Configure the following two tags for an object named exampleobject.png in a bucket named examplebucket: tagkey1#tagvalue1 and tagkey2#tagvalue2.

  ```
  ./ossutil64 object-tagging --method put oss://examplebucket/exampleobject.png.txt tagkey1#tagvalue1 tagkey2#tagvalue2
  ```

- Configure the following three tags for objects whose names contain the "test" prefix in a bucket named examplebucket: tagkey3#tagvalue3, tagkey4#tagvalue4, and tagkey5#tagvalue5.

  ```
  ./ossutil64 object-tagging --method put oss://examplebucket/test -r tagkey3#tagvalue3 tagkey4#tagvalue4 tagkey5#tagvalue5
  ```

- Configure a tag whose key is tagkey6 and whose value is tagvalue6 for the specified version of an object named exampleobject.txt in a bucket named examplebucket.

  ```
  ./ossutil64 object-tagging --method put oss://examplebucket/exampleobject.txt tagkey6#tagvalue6 --version-id CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3****
  ```

  For more information about how to query all versions of an object, see Is.

- After the preceding sample commands are successful, the following similar output is returned to indicate the time used to configure tagging:

  ```
  0.106852(s) elapsed
  ```

Queries object tags

Only the owner of a bucket and RAM users that have the GetObjectTagging permission can query the tags of objects in the bucket.

The following examples show how to query the tags of objects:

- Query the tags of an object

  Query the tags of an object named exampleobject.txt in a bucket named examplebucket.

  ```
  ./ossutil64 object-tagging --method get oss://examplebucket/exampleobject.txt
  ```

  The following output result shows that exampleobject.txt has a tag whose key is tagkey and whose value is tagvalue.

  ```
  object index   tag index      tag key   tag value       object
  ------------------------------------------------------------------------------
  1              0              "tagkey"  "tagvalue"      oss://examplebucket/exampleobject
  .txt
  0.068156(s) elapsed
  ```

- Queries the tags of multiple objects

  Query the tags of all objects whose names contain the "test" prefix in a bucket named examplebucket.

  ```
  ./ossutil64 object-tagging --method get oss://examplebucket/test -r
  ```

  The following output result shows that objects whose names contain the "test" prefix have the following three tags: tagkey3#tagvalue3, tagkey4#tagvalue4, and tagkey5#tagvalue5.

  ```
  object index   tag index      tag key   tag value       object
  ------------------------------------------------------------------------------
  1              0              "tagkey3" "tagvalue3"     oss://examplebucket/test
  1              1              "tagkey4" "tagvalue4"     oss://examplebucket/test
  1              2              "tagkey5" "tagvalue5"     oss://examplebucket/test
  0.093040(s) elapsed
  ```

Remove object tags

Only the owner of a bucket and RAM users that have the DeleteObjectTagging permission can remove the tags of objects in the bucket.

The following examples show how to remove the tags of objects:

- Remove the tags of an object

  You can run the following command to remove the tags of an object named exampleobject.txt in a bucket named examplebucket:

  ```
  ./ossutil64 object-tagging --method delete oss://examplebucket/exampleobject.txt
  ```

- Remove the tags of multiple objects

  You can run the following command to remove the tags of all objects whose names contain the "test" prefix in a bucket named examplebucket:

  ```
  ./ossutil64 object-tagging --method delete oss://examplebucket/test -r
  ```

- If the preceding commands are successful, an output similar to the following is returned to indicate

the time used to remove tags:

```
0.148970(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to configure a tag whose key is tagkey7 and whose value is tagvalue7 for an object named example.png in a bucket named testbucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 object-tagging --method put oss://testbucket/exampletest.png tagkey7#tagvalue7
-e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H*
***
```

For more information the mb command, see Common options.

2.3.26. probe

This topic describes how to run the **probe** command to probe access to Object Storage Service (OSS). You can run this command to detect network exceptions that occur between the local client and the OSS server and check upload and download bandwidths and the status of local symbolic links.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Usage notes

You can run the **probe** command to detect network exceptions that occurred between the local client and the OSS server and check the upload and download bandwidths. You can also run this command to check the status of a large number of symbolic links before you upload them.

- Detect network exceptions

  ossutil checks the network between the local client and the OSS server by uploading or downloading objects. To upload or download objects to the specified location when you detect network exceptions, we recommend that you specify the names of the objects. If you need only to detect network exceptions, you can run the command without specifying object names. ossutil generates temporary objects for detection and deletes the objects after the network is detected.

- Check upload and download bandwidths

  When you run this command to check upload and download bandwidths, OSS provides a recommended number of concurrent tasks in uploads and downloads based on the CPU of your local devices and your network bandwidth. You can configure the number of concurrent tasks for ossutil as recommended to maximize the usage of your network bandwidth.

- Check the status of local symbolic links

When you upload a large number of symbolic links, the upload is interrupted when one of the links is abnormal. We recommend that you check the status of local symbolic links before you upload them. If one of the symbolic links is abnormal, fix the issue before you upload the links.

After you run the **probe** command, you can view the procedure and result of each upload or download task.

- Detect network exceptions by uploading or downloading objects
  - A check sign (√) displayed following a step indicates that the step succeeded. A cross sign (×) following a step indicates that the step failed.
  - If the detection succeeds, the information about the upload or download task is displayed as the result, which includes the size of the uploaded or downloaded object, the time used to perform the upload or download task, and the location to which the object is uploaded or downloaded.
  - If the detection fails, ossutil returns the error causes or error codes, which can be used to troubleshoot the errors.

    For more information about error codes, see Error responses.
  - After a detection, a log file whose name is in the following format is generated in the installation directory of ossutil: `logOssProbe + Timestamp when the detection ends.log`. The log file contains the detailed information about the execution of the **probe** command.
- Check a specific item

  When you run the probe command to check the upload and download bandwidths or the status of local symbolic links, ossutil provides the check results and suggestions.

Check the network by uploading an object and generate a report

ossutil checks the network between the local client and a bucket by uploading an object to the bucket.

- Command syntax

```
./ossutil64 probe {--upload [file_name]} {--bucketname bucket_name} [--object object_name
] [--addr domain_name] [--upmode]
```

The following table describes the parameters that you can configure when you run this command to check the network between the local client and a bucket by uploading an object to the bucket.

Parameter	Required	Description
--upload	Yes	Specifies that the network is detected by uploading an object.
file_name	No	The full path of the local file that you want to upload to the bucket. If you leave this parameter unspecified, ossutil generates a temporary object and detects the network by uploading the object.
--bucketname	No	The name of the bucket to which you want to upload the object.

Parameter	Required	Description
--object	No	The name of the object that you want to upload. If you specify this parameter in the command, ossutil stores the uploaded object in the specified bucket. If you do not specify this parameter in the command, ossutil deletes the uploaded object after the network is detected.
--addr	No	The address of the network that you want to check. ossutil runs the ping command to check the network between the local client and the specified address. Default value: `www.aliyun.com`.
--upmode	No	The method used to upload the object. Default value: normal. Valid values: ○ normal: The object is uploaded by using simple upload. ○ append: The object is uploaded by using append upload. ○ multipart: The object is uploaded by using multipart upload.

- Example
  - Check the network by uploading a random object and specifying the address of the network

    In this example, the bucket to which the object is uploaded is named examplebucket and the address of the network to check is `aliyun.com`. Command:

    ```
    ./ossutil64 probe --upload --bucketname examplebucket --add aliyun.com
    ```

    A similar output is displayed:

    ```
    begin parse parameters and prepare file...[✓]
    begin network detection...[✓]
    begin upload file(normal)...[✓]
    ************************  upload result  *************************
    upload file:success
    upload file size:122880(byte)
    upload time consuming:245(ms)
    (only the time consumed by probe command)
    ************************ report log info*************************
    report log file:/root/logOssProbe20201201173031.log
    ```

- Check the network by uploading the specified object in default simple upload mode and delete the uploaded object after the network is detected

  In this example, the file named *example.txt* in the local root directory is uploaded to the bucket named examplebucket. Command:

  ```
  ./ossutil64 probe --upload example.txt --bucketname examplebucket
  ```

  A similar output is displayed:

  ```
  begin parse parameters and prepare file...[✓]
  begin network detection...[✓]
  begin upload file(normal)...[✓]
  ************************  upload result  *************************
  upload file:success
  upload file size:7(byte)
  upload time consuming:224(ms)
  (only the time consumed by probe command)
  upload object is example.txt
  ************************ report log info*************************
  report log file:/root/logOssProbe20201201173841.log
  ```

- Check the network by uploading the specified object in append upload mode and store the uploaded object in the bucket after detection

  In this example, the file named *example.txt* in the local root directory is uploaded to the bucket named examplebucket. Command:

  ```
  ./ossutil64 probe --upload example.txt --bucketname examplebucket --object example.txt
  --upmode append
  ```

  A similar output is displayed:

  ```
  begin parse parameters and prepare file...[✓]
  begin network detection...[✓]
  begin upload file(append)...[✓]
  ************************  upload result  *************************
  upload file:success
  upload file size:7(byte)
  upload time consuming:171(ms)
  (only the time consumed by probe command)
  upload object is example.txt
  ************************ report log info*************************
  report log file:/root/logOssProbe20201201174126.log
  ```

Check the network by downloading an object using its URL and generate a report

ossutil checks the network between the local client and a bucket by downloading an object using its URL from the bucket.

- Command syntax

  ```
  ./ossutil64 probe {--download} {--url http_url} [--addr=domain_name] [file_name]
  ```

  The following table describes the parameters that you can configure when you run this command to

check the network by downloading an object using its URL.

Parameter	Required	Description
--download	Yes	Specifies that the network is detected by downloading an object.
--url	Yes	The URL of the object that you want to download. ossutil uses this URL to download the object to a local directory. ◦ If the ACL of the object to download is public read, set this parameter to the URL of the object. Example: `https://examplebucket.oss-cn-beijing.aliyuncs.com/example.jpg`. ◦ If the ACL of the object to download is private, set this parameter to the signed URL of the object. The URL must start and end with a pair of double quotation marks (""). Example: `"https://examplebucket.oss-cn-beijing.aliyuncs.com/example.jpg?Expires=1552015472&OSSAccessKeyId=TMP.******5r9f1FV12y8_Qis6LUVmvoSCUSs7aboCCHtydQ0axN32Sn-UvyY3AAAwLAIUarYNLcO87AKMEcE5O3A******oCFAQuRdZYyVFyqOW8QkGAN-bamUiQ&Signature=bIa4llbMbldrl7rwckr%2FXXvTtxw%3D"`.
--addr	No	The address of the network that you want to check. ossutil runs the ping command to check the network between the local client and the specified address. Default value: `www.aliyun.com`.
file_name	No	The local path to which the object is downloaded. ◦ If you specify only the object name but do not specify a directory, the object is stored by using the specified name in the installation directory of ossutil. ◦ If you specify only a directory but do not specify the object name, the object is stored by using its original name in the specified directory. ◦ If you leave this parameter unspecified, the object is stored with its original name in the installation directory of ossutil.

- Example

- Check the network by downloading the specified object using its URL and rename the object

  In this example, the public read object named `example.txt` in a bucket named examplebucket is downloaded using its URL: `https://examplebucket.oss-cn-beijing.aliyuncs.com/example.txt`. The object is renamed `/localfile/test.txt` after it is downloaded. Command:

```
./ossutil64 probe --download --url https://examplebucket.oss-cn-beijing.aliyuncs.com/ex
ample.txt  /localfile/test.txt
```

  A similar output is displayed:

```
begin parse parameters and prepare object...[✓]
begin network detection...[✓]
begin download file...[✓]
************************  download result  ************************
download file:success
download file size:57374182(byte)
download time consuming:1246(ms)
(only the time consumed by probe command)
download file is /localfile/test.txt
************************ report log info************************
report log file:/root/logOssProbe20201202171639.log
```

- Check the network by downloading the specified object using its URL and specifying the address of the network

  In this example, a public-read object named `example.txt` in a bucket named examplebucket is downloaded using its URL: `https://examplebucket.oss-cn-beijing.aliyuncs.com/example.txt`. The address of the network to check is `aliyun.com`. Command:

```
./ossutil64 probe --download --url https://examplebucket.oss-cn-beijing.aliyuncs.com/ex
ample.txt --add aliyun.com
```

  A similar output is displayed:

```
begin parse parameters and prepare object...[✓]
begin network detection...[✓]
begin download file...[✓]
************************  download result  ************************
download file:success
download file size:57374182(byte)
download time consuming:1344(ms)
(only the time consumed by probe command)
download file is /root/example.txt
************************ report log info************************
report log file:/root/logOssProbe20201202172232.log
```

Check the network by downloading the specified object and generate a report

ossutil checks the network between the local client and a bucket by downloading an object from the bucket.

- Command syntax

```
./ossutil64 probe {--download} {--bucketname bucket_name} [--object object_name] [--add domain_name] [file_name]
```

The following table describes the parameters that you can configure when you run this command to check the network between the local client and a bucket by downloading an object from the bucket.

Parameter	Required	Description
--download	Yes	Specifies that the network is detected by downloading an object.
--bucketname	Yes	The name of the bucket from which you want to download the object.
--object	No	The name of the object that you want to download. If you specify this parameter in the command, ossutil downloads the specified object to a local directory. If you do not specify this parameter in the command, ossutil generates a temporary object, uploads the object to the specified bucket, and then downloads the object. After the network is detected, ossutil deletes the temporary object from the specified bucket.
--addr	No	The address of the network that you want to check. ossutil runs the ping command to check the network between the local client and the specified address. Default value: `www.aliyun.com`.
file_name	No	The local path to which the object is downloaded. ◦ If you specify only the object name but do not specify a directory, the object is stored by using the specified name in the installation directory of ossutil. ◦ If you specify only a directory but do not specify the object name, the object is stored by using its original name in the specified directory. ◦ If you leave this parameter unspecified, the object is stored by using its original name in the installation directory of ossutil.

- Examples

- Check the network by downloading the specified object and rename the object

  In this example, the `/ossfolder/example.txt` object in the bucket named examplebucket is downloaded to a local directory and is renamed `/localfolder/test.txt` after the object is downloaded. You can run the following command to check the network:

  ```
  ./ossutil64 probe --download --bucketname examplebucket --object /ossfolder/example.txt
  /localfolder/text.txt
  ```

  A similar output is displayed:

  ```
  begin parse parameters and prepare object...[✓]
  begin network detection...[✓]
  begin download file...[✓]
  ************************  download result  ************************
  download file:success
  download file size:57374182(byte)
  download time consuming:1108(ms)
  (only the time consumed by probe command)
  download file is /localfolder/text.txt
  ************************ report log info************************
  report log file:/root/logOssProbe20201202173032.log
  ```

- Check the network by downloading a temporary object and specifying the address of the network

  In this example, a temporary object is downloaded from the examplebucket bucket. The address of the network to check is `aliyun.com`. Command:

  ```
  ./ossutil64 probe --download --bucketname examplebucket --add aliyun.com
  ```

  A similar output is displayed:

  ```
  begin parse parameters and prepare object...[✓]
  begin network detection...[✓]
  begin download file...[✓]
  ************************  download result  ************************
  download file:success
  download file size:122880(byte)
  download time consuming:516(ms)
  (only the time consumed by probe command)
  download file is /root/oss-test-probe-1606902911701892000-3ifwj9t0ln
  ************************ report log info************************
  report log file:/root/logOssProbe20201202173512.log
  ```

Check a specific index

You can run the probe command to check a specific index, including the status of local symbolic links and the upload and download bandwidths. ossutil also returns the names of abnormal symbolic links or a recommended number of concurrent tasks in uploads and downloads based on the check results.

- Command syntax

  ```
  ./ossutil64 probe {--probe-item item_value} {--bucketname bucket-name} [--object object_n
  ame]
  ```

The following table describes the parameters that you can configure when you run this command to check a specific index.

Parameter	Required	Description
--probe-item	Yes	The index that you want to check. Valid values: ○ cycle-symlink: ossutil checks whether local abnormal symbolic links exist. ○ upload-speed: ossutil checks the upload bandwidth. ○ download-speed: ossutil checks the download bandwidth.
--bucketname	Yes when the value of --probe-item is not `cycle-symlink`	The name of the bucket for which you want to check the upload or download bandwidth.
--object	Yes when the value of --probe-item is `download-speed`	The path of the object that you want to download. The object must exist and be larger than 5 MB in size. Example: `ossfolder/example.txt`.

- Example
  - Check whether abnormal symbolic links exist in the localfolder directory of the local root directory

    Command:

    ```
    ./ossutil64 probe --probe-item cycle-symlink /root/localfolder
    ```

    A similar output is displayed:

    ```
    Error: stat /root/localfolder/example.jpg: no such file or directory
    ```

    The results show that the *example.jpg* symbolic link is abnormal.

- Check the upload bandwidth

  In this example, ossutil uploads a temporary object to a bucket named examplebucket and returns a recommended number of concurrent upload tasks based on the hardware specification of the current device and the upload bandwidth.

  Command:

  ```
  ./ossutil64 probe --probe-item upload-speed --bucketname examplebucket
  ```

  A similar output is displayed:

  ```
  cpu core count:2
  parallel:2,average speed:679.72(KB/s),current speed:1344.00(KB/s),max speed:1440.00(KB/
  s))
  parallel:3,average speed:643.31(KB/s),current speed:704.00(KB/s),max speed:1632.00(KB/s
  ))
  parallel:4,average speed:646.62(KB/s),current speed:512.00(KB/s),max speed:1600.00(KB/s
  ))
  suggest parallel is 2, max average speed is 679.72(KB/s)
  ```

  The results show that the CPU of the device has two cores and the maximum average upload bandwidth is 679.72 KB/s. We recommend that you set the number of concurrent upload tasks to 2 based on these results.

- Check the download bandwidth

  In this example, ossutil downloads an object named `example.txt` from a bucket named examplebucket to a local directory and returns a recommended number of concurrent tasks in download based on the hardware specification of the current device and the download bandwidth.

  Command:

  ```
  ./ossutil64 probe --probe-item download-speed --bucketname examplebucket --object examp
  le.txt
  ```

  A similar output is displayed:

  ```
  cpu core count:2
  parallel:2,average speed:12524.93(KB/s),current speed:12288.63(KB/s),max speed:14302.25
  (KB/s)
  parallel:3,average speed:12564.45(KB/s),current speed:12144.39(KB/s),max speed:14484.24
  (KB/s)
  parallel:4,average speed:12545.21(KB/s),current speed:12766.58(KB/s),max speed:13534.42
  (KB/s)
  suggest parallel is 3, max average speed is 12564.45(KB/s)
  ```

  The results show that the CPU of the device has two cores and the maximum average download bandwidth is 12,564.45 KB/s. We recommend that you set the number of concurrent download tasks to 3 based on these results.

2.3.27. read-symlink (query a symbolic link)

The **read-symlink** command is used to query the information about a symbolic link, including the ETag and last update time. To run this command to query the information about a symbolic link, you must have read permissions on the symbolic link.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 read-symlink oss://bucketname/objectname [--encoding-type <value>] [--payer <va
lue>]
```

The following table describes the parameters that you can configure in this command.

Parameter	Description
bucketname	The name of the bucket in which the symbolic link you want to query is stored.
objectname	The name of the symbolic link you want to query.
--encoding-type	The method used to encode the name of the symbolic link. Valid value: *url*. If you do not specify this parameter, the name of the symbolic link is not encoded.
--payer	The payer of the traffic and request fees incurred when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fee incurred during queries, set this parameter to *requester*.

Examples

Query the information about a symbolic link named test.jpg in a bucket named examplebucket.

```
./ossutil64 read-symlink oss://examplebucket/test.jpg
```

The following output result indicates that the information about test.jpg is obtained, including the ETag value, last update time, and the object to which the symbolic link points. According to the result, test.jpg points to an object named example.jpg.

```
Etag                    : 938F26218CE422CBEEE0B6543A2B2D
Last-Modified           : 2021-04-21 18:00:13 +0800 CST
X-Oss-Symlink-Target    : example.jpg
0.217317(s) elapsed
```

If the object that you query is not a symbolic link, `NotSymlink` is returned.

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by multiple Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to query the information about a symbolic link named testobject.png in a bucket named testbucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 read-symlink oss://testbucket/testobject.png -e oss-cn-shanghai.aliyuncs.com -i
LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options, see Common options.

2.3.28. referer

Object Storage Service (OSS) allows you to configure a Referer whitelist for a bucket to prevent your resources in the bucket from unauthorized access. This topic describes how to run the **referer** command to add, modify, query, or delete hotlink protection configurations for a bucket.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about hotlink protection, see Configure hotlink protection in OSS Developer Guide.

Add or modify hotlink protection configurations for a bucket

If hotlink protection is not configured for a bucket, you can run this command to add hotlink protection configurations for the bucket. If hotlink protection is configured for a bucket, the existing hotlink protection configurations for the bucket are overwritten when you run this command.

You can refer to the following command format and examples to add or modify hotlink protection configurations for a bucket:

- Command syntax

  ```
  ./ossutil64 referer --method put oss://bucketname refererconfig [--disable-empty-referer]
  ```

  The following table describes the parameters that you can configure when you run this command to add or modify hotlink protection configurations.

Parameter	Description
bucketname	The name of the bucket for which you want to add or modify hotlink protection configurations.

Parameter	Description
refererconfig	Specifies the domain name or IP address of the origins from which you want to allow requests. Asterisks (*) and question marks (?) are supported as wildcards. Separate multiple Referer configurations with spaces. The following examples show how to configure domain names and IP addresses: ◦ If you add `www.aliyun.com` to the Referer whitelist, requests sent from URLs that start with *www.aliyun.com*, such as *www.aliyun.com/123* and *www.aliyun.com.cn* are allowed. ◦ If you add `*www.aliyun.com/` to the Referer whitelist, requests sent from *http://www.aliyun.com/* and *https://www.aliyun.com/* are allowed. ◦ You can use an asterisk (*) as a wildcard to indicate zero or more characters. For example, if you add `*.aliyun.com` to the Referer whitelist, requests sent from URLs such as *help.aliyun.com* and *www.aliyun.com* are allowed. ◦ You can use a question mark (?) as a wildcard to indicate a single character. ◦ You can add domain names or IP addresses that include a port number, such as *www.example.com:8080* and *10.0.0.0:8080*, to the Referer whitelist.
--disable-empty-referer	Specifies whether empty Referer is allowed: ◦ If you do not add this parameter, the Referer field is allowed to be empty. HTTP or HTTPS requests that contain an empty Referer field or no Referer field are allowed. ◦ If you add this parameter, the Referer field is not allowed to be empty. Only HTTP or HTTPS requests that include the Referer field can access the bucket.

- Examples

  Configure hotlink protection for a bucket named examplebucket. Only HTTP or HTTPS requests that contain `*www.aliyun.com` in their headers are allowed to access the examplebucket bucket, and the Referer field cannot be empty.

```
./ossutil64 referer --method put oss://examplebucket *www.aliyun.com --disable-empty-refe
rer
```

  Configure hotlink protection for the bucket named examplebucket. Only HTTP or HTTPS requests that contain `example.aliyundoc.com` in their headers are allowed to access the examplebucket bucket, and the Referer field is allowed to be empty.

```
./ossutil64 referer --method put oss://examplebucket example.aliyundoc.com
```

  If a similar output is displayed, hotlink protection configurations are added:

```
0.134839(s) elapsed
```

Query the hotlink protection configurations of a bucket

- Command syntax

```
./ossutil64 referer --method get oss://bucketname [local_xml_file]
```

The following table describes the parameters that you can configure when you run this command to query the hotlink protection configurations of a bucket.

Parameter	Description
bucketname	The name of the bucket of which the hotlink protection configurations you want to query.
local_xml_file	The name of the local file used to store the hotlink protection configurations. Example: `localfile.txt`. If this parameter is not specified, obtained hotlink protection configurations are displayed without being stored in a local file.

- Examples
  - Obtain the hotlink protection configurations of the examplebucket bucket and write the obtained configurations to the `localfile.txt` local file.

```
./ossutil64 referer --method get oss://examplebucket localfile.txt
```

If a similar output is displayed, the hotlink protection configurations are obtained:

```
0.212407(s) elapsed
```

  - Obtain the hotlink protection configurations of the examplebucket bucket and display the configuration result without storing the result in a local file.

```
./ossutil64 referer --method get oss://examplebucket
```

If a similar output is displayed, only requests that contain `*www.aliyun.com` in the HTTP or HTTPS headers are allowed to access the examplebucket bucket based on the hotlink protection configurations of the examplebucket bucket, and Referer must not be empty:

```
<?xml version="1.0" encoding="UTF-8"?>
  <RefererConfiguration>
      <AllowEmptyReferer>false</AllowEmptyReferer>
      <RefererList>
          <Referer>*www.aliyun.com</Referer>
      </RefererList>
  </RefererConfiguration>
0.080482(s) elapsed
```

Delete the hotlink protection configurations of a bucket

- Command syntax

```
./ossutil64 referer --method delete oss://bucketname
```

bucketname specifies the name of the bucket of which the hotlink protection configurations are to be deleted.

- Examples

Delete the hotlink protection configurations for the examplebucket bucket.

```
./ossutil64 referer --method delete oss://examplebucket
```

If a similar output is displayed, the hotlink protection configurations of the bucket are deleted:

```
0.212409(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by multiple Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

The following command provides an example on how to configure hotlink protection for the testbucket bucket owned by another Alibaba Cloud account in the China (Hangzhou) region:

```
./ossutil64 referer --method put oss://testbucket www.alibabacloud.com -e oss-cn-hangzhou.a
liyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the referer command, see Common options.

2.3.29. replication

Cross-region replication (CRR) provides automatic and asynchronous (near real-time) replication of objects across buckets in different Object Storage Service (OSS) regions. This topic describes how to run the **replication** command to manage the configurations of CRR rules. After you configure a CRR rule, OSS replicates the objects in the source bucket to the destination bucket located in a different OSS region based on the CRR rule.

> **Notice**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about the scenarios and usage notes of CRR, see Cross-region replication.

Add a CRR rule

You can configure CRR rules to replicate data from the source bucket to multiple destination buckets. You can configure up to 100 CRR rules for a bucket.

To add a CRR rule, you must first create a local file and specify the CRR rule that you want to add. Then, add the CRR rule for the source bucket.

- Command syntax

```
./ossutil64 replication --method put oss://bucketname local_xml_file
```

The following table describes the parameters that you can configure when you run this command to add CRR rules for a bucket.

Parameter	Description
bucketname	The name of the bucket for which you want to add CRR rules.
local_xml_file	The name of the local file in which the CRR rules are specified. Example: `local file.txt`.

- Examples
  i. Create a file named `localfile.txt` on the local device and specify CRR rules in the file based on your requirements.

     The following code provides an example on how to configure a rule. This rule allows you to synchronize create, delete, and modify operations performed on objects whose names contain the specified srcdir prefix in the source bucket to the destination bucket named destbucket that is located in the China (Hangzhou) region. In addition, transfer acceleration is enabled in the synchronization process, and historical data is also synchronized to the destination bucket.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
   <Rule>
        <PrefixSet>
            <Prefix>srcdir</Prefix>
        </PrefixSet>
        <Action>ALL,PUT</Action>
        <Destination>
            <Bucket>destbucket</Bucket>
            <Location>oss-cn-hangzhou</Location>
            <TransferType>oss_acc</TransferType>
        </Destination>
        <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
   </Rule>
</ReplicationConfiguration>
```

  ii. Add the CRR rule for the source bucket named srcbucket.

```
./ossutil64 replication --method put oss://srcbucket localfile.txt
```

- Output

  If a similar output is displayed, the CRR rule is added for the srcbucket bucket:

```
0.856895(s) elapsed
```

Query the CRR rules added for a bucket

- Command syntax

```
./ossutil64 replication --method get oss://bucketname
```

- Examples

You can run the following command to query the CRR rules added for the srcbucket bucket and view the results that are displayed on the screen:

```
./ossutil64 replication --method get oss://srcbucket
```

If a similar output is displayed, the CRR rules added for the srcbucket bucket are obtained:

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Rule>
    <ID>37417af4-f2dc-4b24-92d3-82092af6****</ID>
    <Action>ALL</Action>
    <Destination>
      <Bucket>destbucket</Bucket>
      <Location>oss-cn-beijing</Location>
    </Destination>
    <Status>doing</Status>
    <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
  </Rule>
</ReplicationConfiguration>
0.069195(s) elapsed
```

Query the regions where CRR can be performed

- Command syntax

  ```
  ./ossutil64 replication --method get --item location oss://bucketname
  ```

- Examples

  You can run the following command to query to which regions data in the srcbucket bucket can be synchronized:

  ```
  ./ossutil64 replication --method get --item location oss://srcbucket
  ```

  If a similar output is displayed, the regions to which you can synchronize data in the srcbucket bucket are obtained:

```
<?xml version="1.0" ?>
<ReplicationLocation>
  <Location>oss-cn-beijing</Location>
  <Location>oss-cn-qingdao</Location>
  <Location>oss-cn-shenzhen</Location>
  <Location>oss-cn-hongkong</Location>
  <Location>oss-us-west-1</Location>
  <LocationTransferTypeConstraint>
    <LocationTransferType>
      <Location>oss-cn-hongkong</Location>
        <TransferTypes>
          <Type>oss_acc</Type>
        </TransferTypes>
      </LocationTransferType>
      <LocationTransferType>
        <Location>oss-us-west-1</Location>
        <TransferTypes>
          <Type>oss_acc</Type>
        </TransferTypes>
      </LocationTransferType>
    </LocationTransferTypeConstraint>
  </ReplicationLocation>
0.226523(s) elapsed
```

Query the progress of a CRR task performed on a specified bucket

- Command syntax

```
./ossutil64 replication --method get --item progress oss://bucketname [ruleID]
```

The following table describes the parameters that you can configure when you run this command to query the progress of a CRR task performed on a specified bucket.

Parameter	Description
bucketname	The name of the bucket on which the CRR task is performed.
ruleID	Specify the ID of the CRR rule of which you want to query the CRR task progress. If you do not specify the ID of a CRR rule, the CRR task progress of all the CRR rules added for the bucket is queried. The results of the CRR task progress can only be displayed on the screen.

- Examples

  You can run the following command to query the progress of a specified CRR task performed on the srcbucket bucket:

```
./ossutil64 replication --method get --item progress oss://srcbucket 37417af4-f2dc-4b24-92d3-82092af6****
```

  You can run the following command to query the progress of all CRR tasks performed on the srcbucket bucket:

```
./ossutil64 replication --method get --item progress oss://srcbucket
```

If a similar output is displayed, the progress of replicating existing data from the srcbucket bucket to the destination bucket before CRR is enabled is 100%. After CRR is enabled, data written to the srcbucket bucket before 06:00:59.000 on August 9, 2021 UTC is replicated to the destination bucket named destbucket.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationProgress>
  <Rule>
    <ID>37417af4-f2dc-4b24-92d3-82092af6****</ID>
    <Action>ALL</Action>
    <Destination>
      <Bucket>destbucket</Bucket>
      <Location>oss-cn-beijing</Location>
    </Destination>
    <Status>doing</Status>
    <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
    <Progress>
      <HistoricalObject>1.00</HistoricalObject>
      <NewObject>2021-08-09T06:00:59.000Z</NewObject>
    </Progress>
  </Rule>
</ReplicationProgress>
0.125002(s) elapsed
```

Remove CRR configurations

- Command syntax

```
./ossutil64 replication --method delete oss://bucketname ruleID
```

- Examples

  You can run the following command to remove the configurations of a specified CRR rule added for the srcbucket bucket:

```
./ossutil64 replication --method delete oss://srcbucket 37417af4-f2dc-4b24-92d3-82092af6*
***
```

  If a similar output is displayed, the configurations of the specified CRR rule added for the srcbucket bucket are removed:

```
0.069195(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of a specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of a specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to configure CRR rules for a bucket named testbucket, which is located in the China (Shanghai) region and is owned by another Alibaba Cloud account. The CRR rule is specified by using the local file named localfile.txt.

```
./ossutil64 replication --method put oss://testbucket localfile.txt -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the replication command, see Common options.

2.3.30. request-payment (configure pay-by-requester)

When pay-by-requester is enabled for a bucket, requesters pay the request and traffic fees that are incurred when the requesters access objects in the bucket. The bucket owner must still pay the storage fees of the objects in the bucket. If you want to share your data without having to pay for additional fees, you can use the **request-payment** command to enable pay-by-requester for your bucket.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

For more information about pay-by-requester, see Enable pay-by-requester.

Enable pay-by-requester

- Command syntax

  ```
  ./ossutil64 request-payment --method put oss://bucketname payment
  ```

  The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket for which you want to enable pay-by-requester.

Parameter	Description
payment	The payment method for the fees that are incurred when third-party users access data in the bucket. Valid values: ○ *Requester*: Requesters are charged for the fee to access objects in the bucket. Anonymous users cannot access a bucket that has pay-by-requester enabled. Requesters must provide authentication information so that OSS can identify and charge requesters for request and traffic fees. If a requester assumes a RAM role of an Alibaba Cloud account to request data, OSS charges the Alibaba Cloud account for the requests sent by the requester and the generated traffic. ○ *BucketOwner*: The bucket owner is charged for the fees when requesters access objects in the bucket.

- Examples

  You can run the following command to enable pay-by-requester for a bucket named examplebucket:

  ```
  ./ossutil64 request-payment --method put oss://examplebucket Requester
  ```

  You can run the following command to disable pay-by-requester for a bucket named examplebucket:

  ```
  ./ossutil64 request-payment --method put oss://examplebucket BucketOwner
  ```

  If a similar output is displayed, pay-by-requester is enabled or disabled for the bucket.

  ```
  0.106852(s) elapsed
  ```

Query the pay-by-requester status of a bucket

- Command syntax

  ```
  ./ossutil64 request-payment --method get oss://bucketname
  ```

  bucketname indicates the bucket for which you want to query the pay-by-requester status.

- Examples

  You can run the following command to query the pay-by-requester status of a bucket named examplebucket:

  ```
  ./ossutil64 request-payment --method get oss://examplebucket
  ```

The following result shows that pay-by-requester is enabled for examplebucket:

```
Requester
0.072024(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to enable pay-by-requester for a bucket named testbucket that is located in the China (Shanghai) region and owned by another Alibaba Cloud account.

```
./ossutil64 request-payment --method put oss://testbucket -e oss-cn-shanghai.aliyuncs.com -
i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information the mb command, see Common options.

2.3.31. restore (restore objects)

Before you access Archive or Cold Archive objects, you must run the **restore** command to restore the objects.

Notice

- Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
- For more information about the status of Archive or Cold Archive objects during restoration and how you are charged for object restoration, see Restore objects.

Command syntax

```
./ossutil64 restore oss://bucketname[/prefix][local_xml_file]
[--encoding-type <value>]
[--payer <value>]
[--version-id <value>]
[-r, --recursive]
[-f, --force]
[--retry-times <value>]
[-j, --job <value>]
```

The following table describes the parameters that you can configure in this command.

Parameter	Description
bucketname	The name of the bucket in which the objects you want to restore is stored.

Parameter	Description
prefix	The resources in the bucket, such as directories and objects.
local_xml_file	The local XML file used to store parameters that are configured to restore Cold Archive objects.
--encoding-type	The method used to encode the value of the prefix parameter. Valid value: *url*. If this parameter is not specified, the prefix parameter is not encoded.
--payer	The payer of the traffic and request fees incurred when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fee incurred during queries, set this parameter to *requester*.
--version-id	The ID of the version of the object you want to restore. This parameter applies only to objects in buckets for which versioning is enabled or suspended.
-r, --recursive	If you specify this parameter, ossutil restores all objects whose names contain the specified prefix in the bucket. If you do not specify this parameter, ossutil restores only the specified object.
-f, --force	Specifies the command to forcibly run without prompting the user for confirmation.
--retry-times	The number of retries after the command fails to be run. Default value: 10. Valid value: 1 to 500.
-j, --job	The number of concurrent tasks performed across multiple objects. Valid values: 1 to 10000. Default value: 3.

Examples

- Restore an Archive object

  OSS takes 1 minute to restore an Archive object. An object cannot be read while it is being restored.

  By default, a restored object remains in the restored state for one day. If you run the restore command for an object that is already in the restored state, the restored state is prolonged by one day. This way, you can prolong the restored state of an object to up to seven days. After the period, the object becomes the frozen state.

  - You can run the following command to restore an Archive object named exampleobject.txt in a bucket named examplebucket:

    ```
    ./ossutil64 restore oss://examplebucket/exampleobject.txt
    ```

  - You can run the following command with the -r parameter specified to restore all Archive objects whose names contain the dest prefix in a bucket named examplebucket:

    ```
    ./ossutil64 restore oss://examplebucket/dest -r
    ```

- You can run the following command to restore the specified version of an object named exampleobject.txt in a bucket named examplebucket:

```
./ossutil64 restore oss://examplebucket/exampleobject.txt --version-id  CAEQARiBgID8rum
R2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3****
```

For more information about how to query all versions of an object, see Is.

- Restore a Cold Archive object

The following example shows how to restore a Cold Archive object named exampleobject.jpg in a bucket named examplebucket within 1 hour and keep the object in the restored state for three days:

i. Create a local XML file named *config.xml* and configure the following parameters in the file:

```
<RestoreRequest>
    <Days>3</Days>
    <JobParameters>
        <Tier>Expedited</Tier>
    </JobParameters>
</RestoreRequest>
```

The following table describes the parameters you can configure.

Parameter	Description
Days	The duration for which you want to keep the restored Cold Archive object in the restored state. Unit: days. Valid values: 1 to 7.
Tier	The restoration priority of the Cold Archive object. Valid values: ■ *Expedited*: The object is restored within 1 hour. ■ *Standard*: The object is restored in 2 to 5 hours. ■ *Bulk*: The object is restored in 5 to 12 hours.

ii. Run the following command to restore exampleobject.jpg:

```
./ossutil64 restore oss://examplebucket/exampleobject.jpg config.xml
```

Notice The time required to restore an object vary with the object size.

- Output

If the preceding command is successful, an output similar to the following is returned to indicate the time used to restore the object:

```
0.106852(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to restore an object named exampletest.png in a bucket named testbucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 restore oss://testbucket/exampletest.png -e oss-cn-shanghai.aliyuncs.com -i LTA
I4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options, see Common options.

2.3.32. revert-versioning

This topic describes how to run the **revert-versioning** command to recover an object in a versioned bucket to the most recent previous version.

> **Note**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - For more information about versioning, see Overview.
> - For more information about how to remove a delete maker, see Delete marker.

Command syntax

```
./ossutil64 revert-versioning oss://bucketname[/prefix]
[--encoding-type <value>]
[-r, --recursive]
[--start-time <value> ]
[--end-time <value>]
[--include <value>]
[--exclude <value>]
[--payer <value>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket that contains the objects you want to recover .
prefix	The resources in the bucket, such as directories and objects.
--encoding-type	The method used to encode the prefix specified by the object name that follows `oss://bucket_name` in the full object path. Valid value: *url*. If this parameter is not specified, the prefix is not encoded.

Parameter	Description
-r, --recursive	If you specify this parameter in the command, ossutil recovers all objects whose names contain the prefix specified by this parameter to the most recent previous version. If you do not specify this parameter in the command, ossutil recovers only the specified object to the most recent previous version.
--start-time	A UNIX timestamp. If you specify this option in the command, objects deleted before this time are not recovered.
--end-time	A UNIX timestamp. If you specify this option in the command, objects deleted after this time are not recovered.
--include	Specifies that the command applies to all objects that meet specified conditions.
--exclude	Specifies that the command applies to all objects that do not meet specified conditions.
--payer	The payer of the traffic and request fees incurred when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees incurred during queries, set this parameter to *requester*.

Examples

- Examples on how to recover specified objects
  - Recover the deleted exampleobject.jpg object in the examplebucket bucket to the most recent previous version.

    ```
    ./ossutil64 revert-versioning oss://examplebucket/exampleobject.jpg
    ```

  - Recover multiple deleted objects that are stored in the examplebucket bucket and whose names contain the specified destdir prefix to the most recent previous version by combining the *-r* option.

    ```
    ./ossutil64 revert-versioning oss://examplebucket/destdir -r
    ```

  - Recover all deleted objects in the examplebucket bucket to the most recent previous version by combining the *-r* option.

    ```
    ./ossutil64 revert-versioning oss://examplebucket  -r
    ```

  - Recover objects that have been deleted during a specified period of time to the most recent previous version.

    Recover objects that have been deleted between 16:22:58, 16 June 2020, UTC+8 and 16:39:38, 16 June 2020, UTC+8 and were stored in the examplebucket bucket to the most recent previous version by combining the *-r* option.

    ```
    ./ossutil64 revert-versioning oss://examplebucket -r --start-time 1592295778 --end-time
    1592296778
    ```

- Recover all deleted objects that match the specified conditions to the most recent previous version.
  - Recover all the deleted TXT objects in the examplebucket bucket to the most recent previous version by combining the *-r* option.

    ```
    ./ossutil64 revert-versioning oss://examplebucket --include "*.txt" -r
    ```

  - Recover all the deleted objects, except for the JPG objects, to the most recent previous version by combining the *-r* option.

    ```
    ./ossutil64 revert-versioning oss://examplebucket --exclude "*.jpg" -r
    ```

- Returned results

  After the specified objects are recovered, a result that contains the number of recovered objects and the time taken to recover these objects is returned. Example:

  ```
  revert versioning object count is 10, batch list count is 10
  0.066413(s) elapsed
  ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to recover the exampletest.png object in the testbucket bucket to the most recent previous version, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 revert-versioning oss://testbucket/exampletest.png -e oss-cn-shanghai.aliyuncs.
com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the revert-versioning command, see Common options.

2.3.33. rm

This topic describes how to run the `rm` command to delete objects, parts, or buckets that you no longer need to avoid unnecessary charges.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 rm oss://bucketname[/prefix]
[-r, --recursive]
[-b, --bucket]
[-m, --multipart]
[-a, --all-type]
[-f, --force]
[--include <value>]
[--exclude <value>]
[--version-id <value>]
[--all-versions]
[--payer <value>]
[--encoding-type <value>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket in which the objects you want to delete is stored.
prefix	The resources in the bucket whose names contain the specified prefix, such as directories or objects.
-r, --recursive	If you specify this parameter, ossutil deletes all objects whose names contain the specified prefix from the bucket. If you do not specify this parameter, ossutil deletes only specified objects.
-b, --bucket	Specifies the operations performed only when you delete buckets.
-m, --multipart	Specifies the incomplete multipart upload tasks initiated to the bucket.
-a, --all-type	Specifies the objects and incomplete multipart upload tasks initiated to the bucket.
-f, --force	Specifies the command to forcibly run without prompting the user for confirmation.
--include	Specifies that all objects that meet specified conditions are included.
--exclude	Specifies that the command applies to all objects that do not meet specified conditions.
--version-id	The ID of the version of the object you want to delete. This parameter applies only to objects in buckets for which versioning is enabled or suspended.
--all-versions	Specifies all versions of objects. This parameter applies only to objects in buckets for which versioning is enabled or suspended. You can specify only one of the --version-id and --all-versions parameters in a single rm command.

Parameter	Description
--payer	The payer of the traffic and request fees charged when the command is run. If you want that the requester who accesses the resources to be charged for the traffic and request fees, set this parameter to *requester*.
--encoding-type	The method used to encode the prefix specified by the object name that follows `oss://bucket_name` in the full object path. Valid value: *url*. If this parameter is not specified, the prefix is not encoded.

Delete objects

- Examples
  - Delete a single object

    Delete an object named exampleobject.txt from a bucket named examplebucket.

    ```
    ./ossutil64 rm oss://examplebucket/exampleobject.txt
    ```

  - Delete all objects whose names contain the "test" prefix from a bucket named examplebucket.

    ```
    ./ossutil64 rm oss://examplebucket/test -r
    ```

  - Delete all objects whose names contain the ".png" suffix from a bucket named examplebucket.

    ```
    ./ossutil64 rm oss://examplebucket  --include "*.png" -r
    ```

  - Delete objects whose names contain the string "abc" and do not contain the ".jpg" or ".txt" suffix from a bucket named examplebucket.

    ```
    ./ossutil64 rm oss://examplebucket  --include "*abc*" --exclude "*.jpg" --exclude "*.tx
    t" -r
    ```

  - Delete the specified version of an object named exampleobject.txt from a versioned bucket named examplebucket.

    ```
    ./ossutil64 rm oss://examplebucket/exampleobject.txt --version-id  CAEQARiBgID8rumR2hYi
    IGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3****
    ```

    For more information about how to query all versions of an object, see ls.

  - Delete all versions of an object named exampleobject.txt from a versioned bucket named examplebucket.

    ```
    ./ossutil64 rm oss://examplebucket/exampleobject.txt --all-versions
    ```

  - Delete all versions of all objects from a versioned bucket named examplebucket.

    ```
    ./ossutil64 rm oss://examplebucket --all-versions -r
    ```

- Output

  After the specified objects are deleted by running the preceding commands, a similar output is displayed, which indicates the number of deleted objects and the time used to delete the objects:

```
Succeed: Total 8 objects. Removed 8 objects.
0.106852(s) elapsed
```

Delete parts

- Examples
  - Specify the -m parameter in the command to delete parts that are generated by the incomplete multipart upload tasks of an object named exampleobject.txt from a bucket named examplebucket.

    ```
    ./ossutil64 rm -m oss://examplebucket/exampleobject.txt
    ```

  - Specify the -m and -r parameters in the command to recursively delete parts that are generated by the incomplete multipart upload tasks of objects whose names contain the "test" prefix from a bucket named examplebucket.

    ```
    ./ossutil64 rm -m oss://examplebucket/test -r
    Do you really mean to remove recursively multipart uploadIds of oss://examplebucket/tes
    t(y or N)? y
    ```

  - Specify the -a and -r parameters in the command to recursively delete objects whose names contain the "src" prefix and parts that are generated by the incomplete multipart upload tasks of these objects from a bucket name examplebucket.

    ```
    ./ossutil64 rm  oss://examplebucket/src -a -r
    Do you really mean to remove recursively objects and multipart uploadIds of oss://examp
    lebucket/src(y or N)? y
    ```

- Output

  After the specified objects and parts are deleted by running the preceding commands, a similar output is displayed, which indicates the number of deleted objects, the number of multipart upload tasks by which the deleted parts are generated, and the time used to delete the objects and parts:

  ```
  Succeed: Total 1 objects, 3 uploadIds. Removed 1 objects, 3 uploadIds.
  1.922915(s) elapsed
  ```

Delete buckets

- Delete the examplebucket bucket that contains no objects or parts.

  ```
  ./ossutil64 rm  oss://examplebucket -b
  Do you really mean to remove the Bucket: examplebucket(y or N)? y
  ```

  A similar output is displayed, which indicates the name of the deleted bucket and the time used to delete the bucket:

  ```
  Removed Bucket: examplebucket
  2.230745(s) elapsed
  ```

- Delete a bucket named examplebucket and all objects and parts in the bucket.

  Warning If you run the following command, all data in the bucket is deleted and deleted data cannot be recovered. Exercise caution when you run this command.

```
./ossutil64 rm  oss://examplebucket -b -a -r
Do you really mean to remove recursively objects and multipart uploadIds of oss://example
bucket(y or N)? y
Do you really mean to remove the Bucket: examplebucket(y or N)? y
```

A similar output is displayed, which indicates the number of deleted objects, the number of multipart upload tasks by which the deleted parts are generated, the name of the deleted bucket, and the time used to delete the objects, parts, and bucket:

```
Succeed: Total 189 objects, 37 uploadIds. Removed 189 objects, 37 uploadIds.
Removed Bucket: examplebucket
9.184193(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to delete an object named exampletest.png from a bucket named testbucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 rm oss://testbucket/exampletest.png -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2
NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

2.3.34. set-acl (configure or modify object or bucket ACLs)

Access control lists (ACLs) are policies used to manage the access permissions of buckets and objects. You can configure an ACL for a bucket when you create the bucket or for an object after you upload the object to Object Storage Service (OSS). You can also modify the ACLs of objects and buckets at any time. The **set-acl** command is used to configure or modify the ACLs of buckets or objects.

> **Note** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Configure or modify the ACL of a bucket

- Command syntax

```
./ossutil64 set-acl oss://bucketname acl -b [--retry-times <value>]
```

The following table describes the parameters that you can configure when you use this command to configure or modify the ACLs of buckets.

Parameter	Description
bucketname	The name of the bucket for which you want to configure or modify ACL.
acl	The ACL of the bucket. Valid values: ◦ *private*: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. ◦ *public-read*: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. If you set the bucket ACL to this value, data leakage may occur and you may be charged additional fees. Exercise caution when you set the bucket ACL to this value. ◦ *public-read-write*: All users, including anonymous users, can perform read and write operations on the objects in the bucket. If you set the bucket ACL to this value, data leakage may occur and you may be charged additional fees. If a user writes illegal information to your objects, your legitimate interests and rights may be infringed. We recommend that you do not set the bucket ACL to this value except in special scenarios.
-b	If you do not specify this parameter in the command, the ACL specified in the command is the ACL of objects. To use the command to configure bucket ACLs, you must specify this parameter.
--retry-times	The number of retries after the command fails to be run. Default value: 10. Valid values: 1 to 500.

- Examples

  You can run the following command to set the ACL of a bucket named examplebucket to private:

```
./ossutil64 set-acl oss://examplebucket private -b
```

Configure or modify the ACLs of objects

- Command syntax

```
./ossutil64 set-acl oss://bucketname[/prefix]acl
[-r]
[--include <value>]
[--exclude <value>]
[--version-id <value>]
[--job <value>]
[--retry-times <value>]
[--encoding-type <value>]
```

  The following table describes the parameters that you can configure when you use this command to configure or modify the ACLs of objects.

Parameter	Description
bucketname	The name of the bucket in which the objects whose ACLs you want to configure or modify.
prefix	The resources in the bucket, such as directories and objects.
acl	The ACL of the objects. Default value: private. Valid values: ◦ *default*: The ACL of the objects is the same as the ACL of the bucket in which the objects are stored. ◦ *private*: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. ◦ *public-read*: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. If you set the object ACL to this value, data leakage may occur and you may be charged additional fees. Exercise caution when you set the bucket ACL to this value. ◦ *public-read-write*: All users, including anonymous users, can perform read and write operations on the objects in the bucket. If you set the object ACL to this value, data leakage may occur and you may be charged additional fees. If a user writes illegal information to your objects, your legitimate interests and rights may be infringed. We recommend that you do not set the bucket ACL to this value except in special scenarios.
-r	If you specify this parameter in the command, ossutil configures the ACL of all objects whose names contain the prefix specified by the prefix parameter. If you do not specify this parameter in the command, ossutil configures the ACL only of the object specified by cloud_url.
--include	Specifies the command to include all objects that meet the specified conditions.
--exclude	Specifies that the command applies to all objects that do not meet the specified conditions.
--version-id	The version ID of the object for which you want to configure ACL. This parameter applies only to objects in buckets for which versioning is enabled or suspended.
--job	The number of concurrent tasks performed across multiple objects. Valid values: 1 to 10000. Default value: 3.
--retry-times	The number of retries after the command fails to be run. Default value: 10. Valid values: 1 to 500.
--encoding-type	Specifies the method used to encode the prefix specified by the object name that follows `oss://bucket_name` in the full object path. Valid value: *url*. If this parameter is not specified, the prefix is not encoded.

- Examples
  - You can run the following command to set the ACL of an object named exampleobject.txt in a bucket named examplebucket to private:

    ```
    ./ossutil64 set-acl oss://examplebucket/exampleobject.txt private
    ```

  - You can run the following command to set the ACL of a version of an object named exampleobject.txt in a bucket named examplebucket to private. The ID of the version is `CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3****`.

    ```
    ./ossutil64 set-acl oss://examplebucket/exampleobject.txt private --version-id CAEQARiB
    gID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3****
    ```

  - You can run the following command to set the ACL of objects whose names contain the "test" prefix in a bucket named examplebucket to default:

    ```
    ./ossutil64 set-acl oss://examplebucket/test default -r
    ```

  - You can run the following command to set the ACL of objects whose names contain the ".jpg" suffix in a bucket named examplebucket to private:

    ```
    ./ossutil64 set-acl oss://examplebucket private --include "*.jpg" -r
    ```

  - You can run the following command to set the ACL of objects whose names contain the string "abc" and do not contain the ".png" and ".txt" suffixes in a bucket named examplebucket to default:

    ```
    ./ossutil64 set-acl oss://examplebucket default --include "*abc*" --exclude "*.png" --e
    xclude "*.txt" -r
    ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to set the ACL of a bucket named examplebucket to private, which is located in the China (Hangzhou) region and owned by another Alibaba Cloud account.

```
./ossutil64 set-acl oss://testbucket private -b -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2
NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that apply to the set-acl command, see Common options.

2.3.35. set-meta

Object Storage Service (OSS) uses object metadata to describe object attributes. Object metadata includes standard HTTP headers and user metadata. HTTP headers can be used to customize the policies of HTTP requests, and user metadata can be used to identify the purposes or attributes of objects. You can run the **set-meta** command to set, modify, or delete metadata for uploaded objects.

Note

- Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
- After you set or update the object metadata by running the **set-meta** command, you can run the **stat** command to view the object metadata. For more information, see stat.

Command syntax

```
./ossutil64 set-meta oss://bucketname[/prefix]
[header:value#header:value...]
[--update]
[--delete]
[-r, --recursive]
[-f, --force]
[--include <value>]
[--version <value>]
[--exclude <value>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket in which the objects whose metadata you want to set or modify are stored.
prefix	The resources stored in the bucket, such as directories and objects.

Parameter	Description

header:value#header:value...	Identifies object metadata by using key-value pairs. Headers are case-insensitive, whereas values are case-sensitive. If you want to set multiple sets of metadata, separate each set of metadata with number signs (#). Example: `Cache-Control:no-cache#Expires:2022-10-12T00:00:00.000Z`.
	You can configure the following headers:
	`Headers:`
	`Expires(time.RFC3339:2006-01-02T15:04:05Z07:00)`
	`X-Oss-Object-Acl`
	`Origin`
	`X-Oss-Storage-Class`
	`Content-Encoding`
	`Cache-Control`
	`Content-Disposition`
	`Accept-Encoding`
	`X-Oss-Server-Side-Encryption`
	`Content-Type`
	`Headers prefixed with the X-Oss-Meta-`
	For more information about how to use headers, see Manage object metadata.

--update	Updates object metadata. This option can be shortened to *-u* and cannot be used together with *--delete*.
--delete	Deletes object metadata. This option cannot be used together with *--update*.
-r, --recursive	If you specify this parameter, ossutil sets metadata for all objects whose names contain the specified prefix in the bucket. If you do not specify this parameter, ossutil sets metadata only for a specified object.
-f, --force	Specifies the command to forcibly run without prompting the user for confirmation.
--version-id	The specified version of the object for which you want to set metadata. This parameter applies only to objects in buckets for which versioning is enabled or suspended.
--include	Specifies that the command applies to all objects that meet specified conditions.
--exclude	Specifies that the command applies to all objects that do not meet specified conditions.

Examples

In the following examples that do not contain *--update* or *--delete*, if *header:value#header:value...* is not specified, only the metadata that starts with `X-Oss-Meta-` is retained, and the value of the object metadata is not changed. For HTTP headers, ossutil uses the interactive mode to ask you whether to retain these headers.

- Set or update the metadata of a single object
  - You can run the following command to set the access control list (ACL) of an object named exampleobject.txt in a bucket named examplebucket to private:

    ```
    ./ossutil64 set-meta oss://examplebucket/exampleobject.txt X-Oss-Object-Acl:private
    ```

  - You can run the following command to set the storage class of an object named exampleobject.txt in a bucket named examplebucket to Standard:

    ```
    ./ossutil64 set-meta oss://examplebucket/exampleobject.txt X-Oss-Storage-Class:Standard
    --version-id  CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3MDRk
    ```

    For more information about how to query the versions of an object, see Is.

  - You can run the following command to set the ACL of an object named exampleobject.txt in a bucket named examplebucket to public-read:

    ```
    ./ossutil64 set-meta oss://examplebucket/exampleobject.txt X-Oss-Object-Acl:public-read
    --update
    ```

    When you use the *--update* option, only the metadata that matches the specified header of the destination object is updated, and the value of the object header is replaced with that of the specified header. You can leave the value of the specified header empty. If you leave the value empty, the value of the header remains unchanged. In the preceding example, the ACL of the exampleobject.txt object is updated to public-read, and other metadata of the object remains unchanged.

- Set or update metadata of multiple objects

  If an error occurs in one of the objects when you set or modify the metadata of multiple objects at the same time, ossutil records the error information of the object in the report object of the ossutil_output directory. Object information of a successful operation is not recorded in the report object.

  - Set object metadata that matches a specified prefix

    The following command provides an example on how to modify the object metadata to change the cache behavior of the objects to *no-cache* and the ACL of the objects to *private* by using the *-r* option. The names of the objects contain the src prefix, and the objects are stored in the examplebucket bucket.

    ```
    ./ossutil64 set-meta oss://examplebucket/src Cache-Control:no-cache#X-Oss-Object-Acl:pr
    ivate -r
    ```

- Modify object metadata that meets specified conditions
  - The following command provides an example on how to modify the object metadata to change the storage class of the objects to Infrequent Access (IA) by using the *-r* and *-u* options. The type of the objects is *.jpg*, and the objects are stored in the desfolder directory of the examplebucket bucket.

    ```
    ./ossutil64 set-meta oss://examplebucket/desfolder/ X-Oss-Storage-Class:IA --include "*.jpg" -u -r
    ```

  - The following command provides an example on how to modify the object metadata to change the storage class of the objects to Archive by using the *-r* and *-u* options. The names of the objects contain *abc*. The types of the objects are not *.jpg* or *.txt*, and the objects are stored in the desfolder directory of the examplebucket bucket.

    ```
    ./ossutil64 set-meta oss://examplebucket/desfolder/ X-Oss-Storage-Class:Archive --include "*abc*" --exclude "*.jpg" --exclude "*.txt" -u -r
    ```

- Delete the user metadata of a specified object

  You can add *--delete* to the set-meta command to delete the user metadata that starts with `X-Oss-Meta-` and whose value is empty for the exampleobject.txt object. The object is in the examplebucket bucket. The object metadata that does not start with `X-Oss-Meta-` is not deleted after you run the following command:

  ```
  ./ossutil64 set-meta oss://examplebucket/exampleobject.txt X-Oss-Meta-Createdby  --delete
  ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to set the metadata of the testobject.jpg object stored in the examplebucket bucket which is located in the China (Shanghai) region and is owned by another Alibaba Cloud account:

```
./ossutil64 set-meta oss://testbucket/testobject.jpg X-Oss-Object-Acl:private  -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the command, see Common options.

2.3.36. sign (generate signed object URLs)

After you upload an object to a bucket, you can generate a signed URL for the object and share the URL with third parties for downloads and previews. This topic describes how to run the **sign** command to generate a signed URL for an object.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 sign cloud_url
[--timeout <value>]
[--version-id <value>]
[--trafic-limit <value>]
[--disable-encode-slash]
[--payer <value>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
cloud_url	The full path of the bucket in which the object is stored.
--timeout	The validity period of the signed URL. Unit: seconds. Valid values: 0 to 9223372036854775807. Default value: 60.
--version-id	The version ID of the object for which you want to generate a signed URL. This parameter applies only to objects in buckets for which versioning is enabled or suspended.
--trafic-limit	The maximum access speed when you use the signed URL and send an HTTP request to access the object. Unit: bit/s. The default value of this parameter is 0, which indicates that the access speed is not limited. Valid values: 819200 to 838860800 (100 KB/s to 100 MB/s).
--disable-encode-slash	Specifies that forward slashes (/) contained in the value of cloud_url are not encoded.
--payer	The payer of the traffic and request fees charged when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees incurred during queries, set this parameter to *requester*.

Examples

- You can run the following command to generate a URL for an object named exampleobject.png in a bucket named examplebucket. The validity period of the URL is the default value, which is 60 seconds.

  ```
  ./ossutil64 sign oss://examplebucket/exampleobject.png
  ```

- You can run the following command to generate a URL for an object named exampleobject.png in a bucket named examplebucket, and set the validity period of the URL to 3,600 seconds:

  ```
  ./ossutil64 sign oss://examplebucket/exampleobject.png --timeout 3600
  ```

- You can run the following command to generate a URL for an object named exampleobject.png in a bucket named examplebucket, set the validity period of the URL to 7,200 seconds, and set the maximum access speed to 100 MB/s:

  ```
  ./ossutil64 sign oss://examplebucket/exampleobject.png --timeout 7200 --trafic-limit 838860800
  ```

- You can run the following command to generate a URL for an object named exampleobject.jpg in a versioned bucket named examplebucket, and set the validity period of the URL to 1,800 seconds:

```
./ossutil64 sign oss://examplebucket/exampleobject.jpg --timeout 1800 --version-id  CAEQA
RiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3****
```

- If the preceding commands are successful, a similar output is returned to indicate the time used to generate the URL, the validity period of the URL, and the signature information in the URL:

```
https://examplebucket.ss-cn-hangzhou.aliyuncs.com/exampleobject.png?Expires=1608282224&OS
SAccessKeyId=LTAI4G33piUmgRN1DXx9****&Signature=jo4%2FGykfuc1A4fvyvKRpRyymYH****
0.368676(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to generate a URL for an object named exampletest.jpg in a bucket named testbucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account, and set the validity period of the URL to 3,600 seconds:

```
./ossutil64 sign oss://testbucket/exampletest.jpg --timeout 3600 -e oss-cn-shanghai.aliyunc
s.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options, see Common options.

2.3.37. stat

This topic describes how to run the **stat** command to query the information of buckets or objects. For example, you can run this command to query the storage class of a bucket and the metadata of an object.

> **Notice**
>
> - Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.
> - Only the bucket owner and Resource Access Management (RAM) users that are granted the `oss:GetObject`, `oss:GetObjectAcl`, and `oss:GetBucketInfo` permissions are allowed to view object metadata by running this command.

Command syntax

```
./ossutil64 stat oss://bucketname[/objectname]
[--encoding-type <value>]
[--payer <value>]
[--version-id <value>]
```

The following table describes the parameters that you can configure when you run this command.

Parameter	Description
bucketname	The name of the bucket whose information you want to query.
objectname	The name of the object whose information you want to query.
--encoding-type	The method to encode the object name. Valid value: *url*. If this option is not specified, the object name is not encoded.
--payer	The payer of the traffic and request fees incurred when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees incurred during queries, set this parameter to *requester*.
--version-id	The ID of the version of the object. This parameter applies only to objects in buckets for which versioning is enabled or suspended.

Examples

- Query the information of the examplebucket bucket.

```
./ossutil64 stat oss://examplebucket
```

If a similar output is displayed, the information of the examplebucket bucket is obtained, which includes the region, creation time, and access control list (ACL) of this bucket:

```
Name               : examplebucket
Location           : oss-cn-hangzhou
CreationDate       : 2021-06-30 16:04:41 +0800 CST
ExtranetEndpoint   : oss-cn-hangzhou.aliyuncs.com
IntranetEndpoint   : oss-cn-hangzhou-internal.aliyuncs.com
ACL                : private
Owner              : 148562088256****
StorageClass       : Standard
RedundancyType     : LRS
0.132413(s) elapsed
```

- Query the information of the exampleobject.jpg object stored in the examplebucket bucket.

```
./ossutil64 stat oss:///examplebucket/exampleobject.jpg
```

If a similar output is displayed, the information of the examplebucket object is obtained, which includes the ETag value, last modified time, and metadata such as the ACL and Contend-Type:

```
ACL                    : default
Accept-Ranges          : bytes
Content-Length         : 8746
Content-Md5            : cvc5wcklut76CVoEi2UW****
Content-Type           : image/jpeg
Etag                   : 72F739C1C925BADEFA095A048B65****
Last-Modified          : 2021-07-27 09:53:03 +0800 CST
Owner                  : 148562088256****
X-Oss-Hash-Crc64ecma   : 56037670485008****
X-Oss-Object-Type      : Normal
X-Oss-Storage-Class    : Standard
0.194739(s) elapsed
```

- Query the information of objects whose names contain special characters and that are stored in the examplebucket bucket.

  For example, you want to query the information of the object whose name is *example txt* and that is stored in the examplebucket bucket. ossutil only supports URL encoding for object names. If an object name contains special characters, you can encode these special characters before you use the object name in the command.

  ```
  ./ossutil64 stat oss://examplebucket/%E7%A4%BA%E4%BE%8B.txt --encoding-type url
  ```

  If a similar output is displayed, the information of the `example txt` object is obtained, which includes the ETag value, last modified time, and metadata such as the ACL and Contend-Type:

  ```
  ACL                    : default
  Accept-Ranges          : bytes
  Content-Length         : 113089
  Content-Md5            : mAYgHM5cSkZNBxKDDqqW****
  Content-Type           : image/jpeg
  Etag                   : 9806201CCE5C4A464D0712830EAA****
  Last-Modified          : 2021-07-27 10:08:57 +0800 CST
  Owner                  : 148562088256****
  X-Oss-Hash-Crc64ecma   : 1106061527435521****
  X-Oss-Object-Type      : Normal
  X-Oss-Storage-Class    : Standard
  0.168335(s) elapsed
  ```

- Query the information of the object of a specified version in the examplebucket bucket.

  ```
  ./ossutil64 stat oss://examplebucket/test.jpg --version-id  CAEQFRiBgICw.YSX1xciIDNiNzc3Z
  TdmNTQzOTQ1OTM4MWUwOWI1Y2M1ZTgz****
  ```

  For more information about how to query all versions of an object, see Is.

  If a similar output is displayed, the information of the object of the specified version is obtained, which includes the ETag value, last modified time, version ID, and metadata such as the ACL and Contend-Type:

```
ACL                 : default
Accept-Ranges       : bytes
Content-Length      : 190540
Content-Md5         : aJQnxHQ2b+zW6hh0c+2y****
Content-Type        : image/jpeg
Etag                : 689427C474366FECD6EA187473ED****
Last-Modified       : 2021-07-27 10:19:25 +0800 CST
Owner               : 148562088256****
X-Oss-Hash-Crc64ecma : 1666171672179499****
X-Oss-Object-Type   : Normal
X-Oss-Storage-Class : Standard
X-Oss-Version-Id    : CAEQFRiBgICw.YSX1xciIDNiNzc3ZTdmNTQzOTQ1OTM4MWUwOWI1Y2M1ZTgz****
0.213528(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to query the information of the testbucket bucket that is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 stat oss://testbucket  -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA*
***  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the stat command, see Common options.

2.3.38. sync (synchronize objects)

2.3.38.1. Overview

This topic describes the **sync** command that you can run to synchronize local files to Object Storage Service (OSS), OSS objects to local disks, or objects between OSS paths.

For more information about how to synchronize local files to OSS, see Synchronize local files to OSS.

For more information about how to synchronize OSS objects to local disks, see Synchronize OSS objects to your computer.

For more information about how to synchronize objects between OSS paths, see Synchronize objects between OSS paths.

2.3.38.2. Synchronize local files to OSS

You can run the **sync** command to synchronize local files to Object Storage Service (OSS).

Usage notes

- Binary name

Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

- Number of synchronized files

  When you run the sync command without including the *--delete* option, the number of files that you can synchronize at a time is unlimited. If you include the *--delete* option, you can synchronize up to 1 million files at a time. If the number of files to synchronize exceeds 1 million, the `over max sync numbers 1000000.` error message is returned.

- Differences between the sync and cp commands
  - The sync command synchronizes files in a specified directory by recursively synchronizing all files and subdirectories in the directory. The cp command recursively synchronizes files only when the *-r* option is specified.
  - If you run the sync command to synchronize data to an OSS bucket, you can add the *--delete* option to the command to delete all objects that do not exist in the source from the destination bucket. This way, only the synchronized objects are retained in the destination bucket. The cp command does not support the *--delete* option.
  - The sync command does not support the *--version-id* option. Therefore, the sync command cannot be used to synchronize previous versions in versioned buckets. The cp command supports the *--version-id* option.

  Except for the preceding differences, the sync command can be used in the same way as the cp command. For more information about how to run the cp command, see cp (Upload objects).

Command syntax

```
./ossutil64 sync file_url cloud_url
[-f --force]
[-u --update]
[--delete]
[--enable-symlink-dir]
[--disable-all-symlink]
[--disable-ignore-error]
[--only-current-dir]
[--output-dir <value>]
[--bigfile-threshold <value>]
[--part-size <value>]
[--checkpoint-dir <value>]
[--encoding-type <value>]
[--snapshot-path <value>]
[--include <value>]
[--exclude <value>]
[--meta <value>]
[--acl <value>]
[--maxupspeed <value>]
[--disable-crc64]
[--payer <value>]
[-j, --job <value>]
[--parallel <value>]
[--retry-times <value>]
[--tagging <value>]
```

The following table describes the parameters that you can configure when you run this command to synchronize local files to OSS.

Parameter	Description
file_url	The path of the local files that you want to synchronize to OSS. Examples: `/localfolder/` in Linux and `D:\localfolder\` in Windows.
cloud_url	The path to which you want to synchronize the local files. The path is in the following format: `oss://bucketname/path`. Example: `oss://examplebucket/exampledir/`. If the value of `cloud_url` does not end with a forward slash (/), ossutil automatically adds one to the end of the value.
-f --force	Specifies the operation to forcibly perform without prompting the user for confirmation.
-u, --update	Specifies that ossutil synchronizes files from the source only when the files do not exist in the destination bucket or when the last modified time of the files is later than that of the corresponding objects in the destination bucket.
--delete	Specifies that only the synchronized objects are retained in the destination bucket. Other objects in the destination bucket are deleted. **Warning** Before you add the --delete option to the command, we recommend that you enable versioning for the destination bucket to prevent data from being accidentally deleted. For more information about versioning, see Overview.
--enable-symlink-dir	Specifies whether to synchronize subdirectories to which symbolic links point.
--disable-all-symlink	Specifies that all files and subdirectories to which symbolic links in the directory point are not synchronized.
--disable-ignore-error	Specifies that errors are not ignored during batch operations.
--only-current-dir	Specifies that only files in the current directory are synchronized. Subdirectories in the current directory and files in these subdirectories are not synchronized.
--output-dir	Specifies the directory in which output objects are stored. Output objects are reports that are generated when errors occur during batch synchronization of files. By default, these reports are stored in the *ossutil_output* directory of the current directory.
-bigfile-threshold	Specifies the maximum size of files that can be synchronized by using resumable upload or download. Unit: bytes. Default value: 104857600 (100 MB). Valid values: 0 to 9223372036854775807.

Parameter	Description
--part-size	Specifies the part size. Unit: bytes. By default, ossutil determines the part size based on the object size. Valid values: 1 to 9223372036854775807.
--checkpoint-dir	Specifies the directory in which the log information of resumable upload or download tasks is stored. When a resumable upload or download task fails, ossutil automatically creates the `.ossutil_checkpoint` directory and records the checkpoint information in the directory. After the resumable upload or download task is completed, ossutil deletes this directory. If this option is specified, make sure that you have permissions to delete the specified directory.
--encoding-type	Specifies the method used to encode the names of objects. Valid values: *url*. If you do not specify this parameter, the names of objects are not encoded.
--snapshot-path	Specifies the directory in which the snapshots of synchronized objects are stored. In the next synchronization task, ossutil reads the snapshots in this directory for incremental synchronization.
--include	Specifies that the command applies to all files that meet the specified conditions.
--exclude	Specifies that the command applies to all files that do not meet the specified conditions.
--meta	Specifies that the metadata of an object is in the `header:value#header:value` format. Example: `Cache-Control:no-cache#Content-Encoding:gzip`. For more information about object metadata, see set-meta.

Parameter	Description
--acl	Specifies the access control list (ACL) of the object to download. Default value: private. Valid values: • *default*: The ACL of the objects is the same as the ACL of the bucket in which the objects are stored. • *private*: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. • *public-read*: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. If you set the object ACL to this value, data leakage may occur and you may be charged out-of-control fees. If a user writes illegal information to your objects, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set the object ACL to this value except in special cases. • *public-read-write*: All users, including anonymous users, can perform read and write operations on the objects in the bucket. If you set the object ACL to this value, data leakage may occur and you may be charged out-of-control fees. Exercise caution when you set the bucket ACL to this value.
--maxupspeed	Specifies the maximum upload speed. Unit: KB/s. Default value: 0. The value of 0 indicates that the upload speed is not limited.
--disable-crc64	Disables CRC-64 in synchronization.
--payer	The payer of the traffic and request fees charged when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees incurred during queries, set this parameter to *requester*.
-j, --job	Specifies the number of files to synchronize concurrently. Valid values: 1 to 10000. Default value: 3.
--parallel	Specifies the number of concurrent tasks run to synchronize a single file. Valid values: 1 to 10000. By default, if you do not set this parameter, ossutil sets the value of this parameter based on the operation type and object size.
--retry-times	Specifies the number of retries after an error occurs when the command is run. Default value: 10. Valid values: 1 to 500.
--tagging	Specifies that the tags of objects are in the `TagkeyA=TagvalueA&TagkeyB=TagvalueB....` format.

Examples

Two files named *d.txt* and *e.png* are stored in the *localfolder* directory of the local root directory. The *destfolder* directory of a bucket named examplebucket contains two objects named *a.txt* and *b.txt* and a subdirectory named *C*. The following structure shows the files and directories in your computer and OSS before synchronization:

```
Local root directory                examplebucket
    └── localfolder          └── destfolder/
            ├── d.txt                ├── a.txt
            ├── e.png                ├── b.txt
                                     └── C/
```

The following examples show how to run the sync command to synchronize local files to OSS in different scenarios.

- Synchronize the *localfolder* directory from your computer to the examplebucket in OSS.

  ```
  ./ossutil64 sync localfolder/  oss://examplebucket/destfolder/
  ```

  After you run the preceding command, the *d.txt* and *e.png* objects are added to the *destfolder* directory of examplebucket. The following structure shows the files and directories in your computer and OSS after synchronization:

  ```
  Local root directory                examplebucket
      └── localfolder          └── destfolder/
              ├── d.txt                ├── a.txt
              ├── e.png                ├── b.txt
                                       ├── d.txt
                                       ├── e.png
                                       └── C/
  ```

- Synchronize the localfolder directory from your computer to examplebucket and retain only the synchronized objects in the destfolder directory of examplebucket.

  You can add the --delete option in the command to delete all objects that do not exist in localfolder from the destfolder. This way, only the synchronized objects are retained in destfolder.

  ```
  ./ossutil64 sync localfolder/  oss://examplebucket/destfolder/ --delete
  ```

  After you run the command, the *localfolder* directory is synchronized from your computer to examplebucket. The *a.txt* and *b.txt* objects and the *C* subdirectory within the *destfolder* directory of examplebucket are deleted. Only the synchronized *d.txt* and *e.png* files are retained in the destfolder directory of examplebucket. The following structure shows the files and directories in your computer and OSS after synchronization:

  ```
  Local root directory                examplebucket
      └── localfolder          └── destfolder/
              ├── d.txt                ├── d.txt
              ├── e.png                ├── e.png
  ```

- Synchronize the localfolder from your computer to examplebucket without confirmation.

  By default, when you synchronize a local directory to an OSS bucket, if the OSS bucket contains objects that have the same name as files in the local directory, OSS prompts you to confirm whether you want to overwrite the existing objects in the bucket. To receive confirmation when you synchronize the localfolder directory from your computer to examplebucket, run the following command:

```
./ossutil64 sync localfolder/ oss://examplebucket/destfolder/
cp: overwrite "oss://examplebucket/destfolder/d.txt"(y or N)?
```

If you confirm that the objects in examplebucket can be overwritten, you can add the *-f,--force* option to the command to force the overwrite operation. In this case, run the following command:

```
./ossutil64 sync localfolder/ oss://examplebucket/destfolder/ -f,--force
```

After you run the command, the *d.txt* and *e.png* objects are added to the *destfolder* directory of examplebucket. The following structure shows the files and directories in your computer and OSS after synchronization:

```
Local root directory                    examplebucket
    └── localfolder          └── destfolder/
            ├── d.txt                ├── a.txt
            ├── e.png                ├── b.txt
                                     ├── d.txt
                                     ├── e.png
                                     └── C/
```

- If the preceding commands are successful, a similar output is returned to indicate the number of synchronized objects, the sizes of synchronized objects, and the time used for the synchronization tasks:

```
Succeed: Total num: 2, size: 750,081. OK num: 2(upload 2 files).
average speed 1641000(byte/s)
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of a specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of a specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to synchronize local files in a local directory named srcfolder to the testfolder directory of a bucket named examplebucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account:

```
./ossutil64 sync srcfolder/ oss://examplebucket/testfolder/ -e oss-cn-shanghai.aliyuncs.com
-i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the sync command, see Common options.

2.3.38.3. Synchronize OSS objects to your computer

You can run the **sync** command to synchronize objects stored in Object Storage Service (OSS) to your computer.

Usage notes

- Binary name

Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

- Number of synchronized files

  When you run the sync command without including the *--delete* option, the number of files that you can synchronize at a time is unlimited. If you include the *--delete* option, you can synchronize up to 1 million files at a time. If the number of files to synchronize exceeds 1 million, the `over max sync numbers 1000000.` error message is returned.

- Differences between the sync and cp commands
  - The sync command synchronizes objects in a specified directory by recursively synchronizing all objects and subdirectories in the directory. The cp command recursively synchronizes objects only when the *-r* option is specified.
  - The sync command supports the *--backup-dir* option. When you run the sync command to synchronize data to OSS, you can add this option to the command to specify a directory in OSS to store objects that exist in the destination but do not exist in the source. The cp command does not support the *--backup-dir* option.
  - The sync command does not support the *--version-id* option. Therefore, you cannot run the sync command to synchronize previous versions in versioned buckets. The cp command supports the *--version-id* option.

  Except for the preceding differences, the sync command can be run in the same way as the cp command. For more information about how to run the cp command, see Download objects.

Command syntax

```
./ossutil64 sync cloud_url  file_url
[-f --force]
[-u --update]
[--maxdownspeed <value>]
[--delete]
[--backup-dir <value>]
[--enable-symlink-dir]
[--disable-all-symlink]
[--disable-ignore-error]
[--only-current-dir]
[--output-dir <value>]
[--bigfile-threshold <value>]
[--part-size <value>]
[--checkpoint-dir <value>]
[--range <value>]
[--encoding-type <value>]
[--snapshot-path <value>]
[--include <value>]
[--exclude <value>]
[--disable-crc64]
[--payer <value>]
[-j, --job <value>]
[--parallel <value>]
[--retry-times <value>]
```

The following table describes the parameters that you can configure to run the sync command to synchronize OSS objects to your computer.

Parameter	Description
cloud_url	Specifies the path of the directory in which the objects you want to synchronize are stored. The path is in the following format: `oss://bucketname/path`. Example: `oss://examplebucket/exampledir/`. If the value of `cloud_url` does not end with a forward slash (/), ossutil automatically adds one to the end of the value.
file_url	Specifies the path of the local directory to which you want to synchronize objects. Examples: `/localfolder/` in Linux and `D:\localfolder\` in Windows.
-f --force	Forces an operation without prompting the user for confirmation.
-u, --update	Specifies that ossutil synchronizes objects from the source only when the objects do not exist in the destination or when the last modified time of the object is later than that of the corresponding objects in the destination.
--maxdownspeed	Specifies the maximum download speed. Unit: KB/s. Default value: 0, which indicates that the download speed is unlimited.
--delete	Specifies that only the synchronized objects are retained in the destination local path. Other objects in the destination local path are deleted. **Warning** To add the --delete option to the command, we recommend that you enable versioning for the bucket to prevent data from being accidentally deleted. For more information about versioning, see Overview.
--backup-dir	Specifies the directory used to store objects that exist in the destination local path but do not exist in the source OSS bucket.
--enable-symlink-dir	Specifies whether to synchronize subdirectories to which symbolic links point.
--disable-all-symlink	Specifies that all objects and subdirectories to which symbolic links in the directory point are not synchronized.
--disable-ignore-error	Specifies that errors are not ignored during batch operations.
--only-current-dir	Specifies that only objects in the current directory are synchronized. Subdirectories in the current directory and objects in these subdirectories are not synchronized.

Parameter	Description
--output-dir	Specifies the directory in which output objects are stored. Output objects are reports that are generated when errors occur during batch synchronization of objects. By default, these reports are stored in the *ossutil_output* directory of the current directory.
-bigfile-threshold	Specifies the maximum size of objects that can be synchronized by using resumable upload or download. Unit: byte. Default value: 104857600 (100 MB). Valid values: 0 to 9223372036854775807.
--part-size	Specifies the part size. Unit: byte. By default, ossutil determines the part size based on the object size. Valid values: 1 to 9223372036854775807.
--checkpoint-dir	Specifies the directory in which the log information of resumable upload or download tasks is stored. Default value: `.ossutil_checkpoint`. When a resumable synchronization task fails, ossutil creates this directory and stores the checkpoint information about the task in this directory. ossutil deletes the directory after the task is completed. Therefore, if you specify a checkpoint directory, make sure that ossutil can delete the directory.
--range	Specifies that a specific range of the object is synchronized and stored as a new file in the destination local path. The minimum start value of the range is 0, which indicates the byte 0 of the object. You can set this parameter in one of the following formats: • Data range For example, a value of `3-9` indicates a range from byte 3 to byte 9, which includes byte 3 and byte 9. • Start value For example, a value of `3-` indicates a range from byte 3 to the end of the object, which includes byte 3. • End value For example, a value of `-9` indicates a range from byte 0 to byte 9, which includes byte 9.
--encoding-type	Specifies the method used to encode the names of objects. Valid values: *url*. If you do not specify this parameter, the names of objects are not encoded.
--snapshot-path	Specifies the directory in which the snapshots of synchronized objects are stored. In the next synchronization task, ossutil reads the snapshots in this directory for incremental synchronization.
--include	Specifies that the command applies to all objects that meet the specified conditions.

Parameter	Description
--exclude	Specifies that the command applies to all objects that do not meet the specified conditions.
--disable-crc64	Disables CRC-64 in synchronization.
--payer	Specifies the payer of the traffic and request fees charged when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees charged during queries, set this parameter to *requester*.
-j, --job	Specifies the number of concurrent tasks performed across multiple objects. Valid values: 1 to 10000. Default value: 3.
--parallel	Specifies the number of concurrent tasks performed for a single object. Valid values: 1 to 10000. By default, if you do not set this parameter, ossutil sets the value of this parameter based on the operation type and object size.
--retry-times	Specifies the number of retries after the command fails to be run. Default value: 10. Valid values: 1 to 500.

Examples

The *localdir* directory of a bucket named examplebucket contains two objects named *a.txt* and *b.txt* and a subdirectory named *C*. A local file named *d.txt* is stored in the *destdir* directory of the local root directory. The following structure shows the objects and directories in your computer and OSS before synchronization:

```
examplebucket           Local root directory
└── localdir/             └── destdir/
      ├── a.txt                 └── d.txt
      ├── b.txt
      └── C/
```

- Synchronize the *localdir* directory from examplebucket to your computer.

  ```
  ./ossutil64 sync  oss://examplebucket/localdir/  destdir/
  ```

  After you run the preceding command, the *a.txt* and *b.txt* objects and the *C* directory are added to the *destfolder* directory of your computer. The following structure shows the objects and directories in your computer and OSS after synchronization:

  ```
  examplebucket           Local root directory
  └── localdir/             └── destdir/
        ├── a.txt                 ├── a.txt
        ├── b.txt                 ├── b.txt
        └── C/                    ├── d.txt
                                  └── C/
  ```

- Synchronize the *localdir* directory of examplebucket to a local directory named *destdir*. The *--backup-dir* parameter is set to a directory named *backup* in the command to specify that all objects exist in the destdir directory but do not exist in localdir are deleted from destdir and are saved in the backup

directory.

```
./ossutil64 sync oss://examplebucket/localdir/  destdir/  --delete  --backup-dir backup/
```

After you run the command, the *localdir* directory of examplebucket is synchronized to the destdir directory in your computer. Objects that exist in the *destdir* directory and do not exist in localdir are moved to the *backup* directory. Only the synchronized objects named *a.txt* and *b.txt* and directory named *C* are stored in the *destdir* directory after synchronization. The *d.txt* object that exists in the destdir directory before synchronization is moved to the *backup* directory. The following structure shows the objects and directories in your computer and OSS after synchronization by specifying the backup directory:

```
examplebucket                Local root directory
└── localdir/                 ├── destdir/
       ├── a.txt              │     ├── a.txt
       ├── b.txt              │     ├── b.txt
       └── C/                 │     └── C/
                              │        └── backup/
                                             └──d.txt
```

- If the preceding commands are successful, a similar output is returned to indicate the number of synchronized objects, the sizes of synchronized objects, and the time used for the synchronization tasks:

```
Succeed: Total num: 2, size: 750,081. OK num: 2(upload 2 files).
average speed 1641000(byte/s)
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to synchronize a directory named srcfolder from a bucket named examplebucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account, to a local directory named examplefolder:

```
./ossutil64 sync oss://examplebucket/srcfolder/  examplefolder/ -e oss-cn-shanghai.aliyuncs
.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the sync command, see Common options.

2.3.38.4. Synchronize objects between OSS paths

You can run the **sync** command to synchronize objects between Object Storage Service (OSS) buckets in the same region or between directories of the same bucket.

Usage notes

- Binary name

Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

- Number of synchronized files

  When you run the sync command without including the *--delete* option, the number of files that you can synchronize at a time is unlimited. If you include the *--delete* option, you can synchronize up to 1 million files at a time. If the number of files to synchronize exceeds 1 million, the `over max sync numbers 1000000.` error message is returned.

- Differences between the sync and cp commands
  - The sync command synchronizes objects in a specified directory by recursively synchronizing all objects and subdirectories in the directory. The cp command recursively synchronizes objects only when the *-r* option is specified.
  - If you run the sync command to synchronize data to an OSS bucket, you can add the *--delete* option to the command to delete all objects that do not exist in the source from the destination. In this case, only the synchronized objects are retained in the destination. The cp command does not support the *--delete* option.
  - The sync command does not support the *--version-id* option. Therefore, you cannot run the sync command to synchronize previous versions in versioned buckets. The cp command supports the *--version-id* option.

  Except for the preceding differences, the sync command can be run in the same way as the cp command. For more information about how to run the cp command, see Copy objects.

Command syntax

```
./ossutil64 sync cloud_url cloud_url
[-f --force]
[-u --update]
[--delete]
[--enable-symlink-dir]
[--disable-all-symlink]
[--disable-ignore-error]
[--only-current-dir]
[--output-dir <value>]
[--bigfile-threshold <value>]
[--part-size <value>]
[--checkpoint-dir <value>]
[--encoding-type <value>]
[--include <value>]
[--exclude <value>]
[--meta <value>]
[--acl <value>]
[--maxupspeed <value>]
[--disable-crc64]
[--payer <value>]
[-j, --job <value>]
[--parallel <value>]
[--retry-times <value>]
[--tagging <value>]
```

The following table describes the parameters that you can configure to run the sync command to synchronize objects between OSS paths.

Parameter	Description
cloud_url	Specifies the paths of the source directory and destination directory. The paths are in the following format: `oss://bucketname/path`. For example, to synchronize the srcdir directory of a bucket named examplebucket to the destdir directory of the same bucket, set the source directory to `oss://examplebucket/srcdir/` and the destination directory to `oss://examplebucket/destcdir/`. If the value of `cloud_url` does not end with a forward slash (/), ossutil automatically adds one to the end of the value.
-f --force	Forces an operation without prompting the user for confirmation.
-u, --update	Specifies that ossutil synchronizes objects from the source only when the objects do not exist in the destination or when the last modified time of the object is later than that of the corresponding objects in the destination.
--delete	Specifies that only the synchronized objects are retained in the destination path. Other objects in the destination path are deleted. **Warning** To add the --delete option to the command, we recommend that you enable versioning for the bucket to prevent data from being accidentally deleted. For more information about versioning, see Overview.
--enable-symlink-dir	Specifies whether to synchronize subdirectories to which symbolic links point.
--disable-all-symlink	Specifies that all objects and subdirectories to which symbolic links in the directory point are not synchronized.
--disable-ignore-error	Specifies that errors are not ignored during batch operations.
--only-current-dir	Specifies that only objects in the current directory are synchronized. Subdirectories in the current directory and objects in these subdirectories are not synchronized.
--output-dir	Specifies the directory in which output objects are stored. Output objects are reports that are generated when errors occur during batch synchronization of objects. By default, these reports are stored in the *ossutil_output* directory of the current directory.
-bigfile-threshold	Specifies the maximum size of objects that can be synchronized by using resumable upload or download. Unit: byte. Default value: 104857600 (100 MB). Valid values: 0 to 9223372036854775807.

Parameter	Description
--part-size	Specifies the part size. Unit: byte. By default, ossutil determines the part size based on the object size. Valid values: 1 to 9223372036854775807.
--checkpoint-dir	Specifies the directory in which the log information of resumable upload or download tasks is stored. Default value: `.ossutil_checkpoint`. When a resumable synchronization task fails, ossutil creates this directory and stores the checkpoint information about the task in this directory. ossutil deletes the directory after the task is completed. Therefore, if you specify a checkpoint directory, make sure that ossutil can delete the directory.
--encoding-type	Specifies the method used to encode the names of objects. Set the value to *url*. If you do not specify this parameter, the names of objects are not encoded.
--include	Specifies that the command applies to all objects that meet the specified conditions.
--exclude	Specifies that the command applies to all objects that do not meet the specified conditions.
--meta	Specifies the metadata of an object in the `header:value#header:value` format. Example: `Cache-Control:no-cache#Content-Encoding:gzip`. For more information about object metadata, see set-meta.
--acl	Specifies the access control list (ACL) of an object. Default value: private. Valid values: • *default*: The ACL of the objects is the same as the ACL of the bucket in which the objects are stored. • *private*: Only the bucket owner can perform read and write operations on objects in the bucket. Other users cannot access the objects in the bucket. • *public-read*: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. If you set the object ACL to this value, data leakage may occur and you may be charged additional fees. If a user writes illegal information to your objects, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set the object ACL to this value except in special cases. • *public-read-write*: All users, including anonymous users, can perform read and write operations on the objects in the bucket. If you set the object ACL to this value, data leakage may occur and you may be charged additional fees. Exercise caution when you set the bucket ACL to this value.
--maxupspeed	Specifies the maximum upload speed. Unit: KB/s. Default value: 0. The value of 0 indicates that the upload speed is not limited.

Parameter	Description
--disable-crc64	Disables CRC-64 in synchronization.
--payer	Specifies the payer of the traffic and request fees charged when the command is run. If you want the requester who accesses the resources in the specified path to pay for the traffic and request fees charged during queries, set this parameter to *requester*.
-j, --job	Specifies the number of concurrent tasks performed across multiple objects. Valid values: 1 to 10000. Default value: 3.
--parallel	Specifies the number of concurrent tasks performed for a single object. Valid values: 1 to 10000. By default, if you do not set this parameter, ossutil sets the value of this parameter based on the operation type and object size.
--retry-times	Specifies the number of retries after the command fails to be run. Default value: 10. Valid values: 1 to 500.
--tagging	Specifies the tags of objects in the following format: `TagkeyA=TagvalueA&TagkeyB=TagvalueB....` .

Examples

Your Alibaba Cloud account owns two buckets named examplebucket1 and examplebucket2. The examplebucket1 bucket contains two directories named *example* and *srcdir*. The examplebucket bucket contains a directory named *destdir*. The following structure shows the objects and directories in the two buckets before synchronization:

```
examplebucket1          examplebucket2
├── exampledir/            └── destdir/
│       ├── a.txt              ├── c.txt
│       └── b.txt              └── e.txt
└── srcdir/
        └── d.txt
```

The following examples show how to run the sync command to synchronize objects between OSS paths in different scenarios:

- Synchronize the *exampledir* directory of the examplebucket1 bucket to the *srcdir* directory of the same bucket.

  ```
  ./ossutil64 sync oss://examplebucket1/exampledir/  oss://examplebucket1/srcdir/
  ```

  After you run the preceding command, all objects in the *exampledir* directory are synchronized to the *srcdir* directory. The following structure shows the objects and directories in examplebucket1 after synchronization:

```
examplebucket1                    examplebucket1
├── exampledir/                   ├── exampledir/
│       ├── a.txt                 │       ├── a.txt
│       └── b.txt                 │       └── b.txt
└── srcdir/                       └── srcdir/
        └── d.txt                             ├── a.txt
                                              ├── b.txt
                                              └── d.txt
```

- Synchronize the *exampledir* directory of the examplebucket1 bucket to the *destdir* directory of the examplebucket2 bucket.

  ```
  ./ossutil64 sync oss://examplebucket1/exampledir/ oss://examplebucket2/destdir/
  ```

  After you run the preceding command, all objects in the *exampledir* directory of examplebucket1 are synchronized to the *destdir* directory of examplebucket2. The following structure shows the objects and directories in examplebucket1 and examplebucket2 after synchronization:

  ```
  examplebucket1            examplebucket2
  ├── exampledir/               └── destdir/
  │       ├── a.txt                 ├── a.txt
  │       └── b.txt                 ├── b.txt
  └── srcdir/                       ├── c.txt
          └── d.txt                 └── e.txt
  ```

- Specify the *--delete* parameter to synchronize all objects from examplebucket1 to examplebucket2 and delete objects exist in examplebucket2 but do not exist in examplebucket1.

  ```
  ./ossutil64 sync oss://examplebucket1 oss://examplebucket2 --delete
  ```

  After you run the preceding command, the *exampledir* and *test* directories of examplebucket1 are synchronized to examplebucket2. The *c.txt* and *e.txt* objects that exist in examplebucket2 before synchronization are deleted. The following structure shows the objects and directories in examplebucket1 and examplebucket2 after synchronization:

  ```
  examplebucket1            examplebucket2
  ├── exampledir/               ├── exampledir/
  │       ├── a.txt             │       ├── a.txt
  │       └── b.txt             │       └── b.txt
  └── srcdir/                   └── srcdir/
          └── d.txt                     └── d.txt
  ```

- If the preceding commands are successful, a similar output is returned to indicate the number of synchronized objects, the sizes of synchronized objects, and the time used for the synchronization tasks:

  ```
  Succeed: Total num: 2, size: 750,081. OK num: 2(upload 2 files).
  average speed 1641000(byte/s)
  ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to synchronize a directory named srcfolder from a bucket named examplebucket, which is located in the China (Shanghai) region and owned by another Alibaba Cloud account, to a directory named examplefolder in a bucket named testbucket:

```
./ossutil64 sync oss://examplebucket/srcfolder/  oss://testbucket/examplefolder/ -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the sync command, see Common options.

2.3.39. update

This topic describes how to run the **update** command to update ossutil to the latest version.

> **Notice** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 update [-f]
```

This command is run to check the current and the latest versions of ossutil and generate their version numbers. If an update is available, you are prompted to determine whether to upgrade ossutil to the latest version. If the *-f* option is specified and an update is available, ossutil is upgraded without prompting the user for confirmation after the command is run.

Examples

- You can run the following command to determine whether to upgrade ossutil to the latest version:

  ```
  ./ossutil64 update
  The current version is V1.5.1, and the latest version is V1.7.5.
  Are you sure that you want to upgrade the version (y or N)? y
  ```

- You can run the following command to upgrade ossutil without prompting the user for confirmation:

  ```
  ./ossutil64 update -f
  ```

2.3.40. website

You can use the static website hosting feature to host your static website on an Object Storage Service (OSS) bucket and use the endpoint of the bucket to access the website. After you configure static website hosting for a bucket, if the data that your users request from OSS does not exist, you can configure back-to-origin rules to make sure that they can still obtain the data. This topic describes how to run the **website** command to add, modify, query, or delete static website hosting, and back-to-origin configurations for a bucket.

Usage notes

- Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace

./ossutil64 in the commands with the corresponding binary name. For more information, see ossutil.

- After static website hosting is configured for a bucket, a 404 error is returned when the data requested by your users from OSS does not exist. However, if you configure back-to-origin rules that contain a valid origin for the bucket, your users can obtain the data based on the rules.

  For more information about static website hosting, see Overview.

  For more information about back-to-origin, see Manage back-to-origin configurations.

Add or modify website-related configurations

If a bucket has no website-related configurations, new configurations are added for the bucket when you run the website command. If a bucket already has website-related configurations, new website configurations overwrite the existing ones when you run the website command.

To add or modify website-related configurations, perform the following steps:

1. Create a local file and add website-related configurations to the file.
2. Add website-related configurations to a destination bucket.

The following code provides an example on the command syntax to add or modify website-related configurations:

- Command syntax

  ```
  ./ossutil64 website --method put oss://bucketname local_xml_file
  ```

  The following table describes the parameters that you can configure when you run this command to add or modify website-related configurations.

Parameter	Description
bucketname	The name of the bucket to which website-related configurations are added or modified.
local_xml_file	The name of the local file in which the website rules are configured. Example: `localfile.txt`.

- Examples

  i. Create a file named `localfile.txt` on the local device and configure different website rules in the file.

  The following code provides an example on how to add complete website-related configurations to a bucket, which includes static website hosting configurations and back-to-origin configurations: You can choose to add only static website hosting configurations or back-to-origin configurations based on your actual requirements.

  - Static website hosting configurations

    In the following sample configuration, the <IndexDocument> field specifies the default homepage of the static website as `index.html`. The <ErrorDocument> field specifies the default 404 error page of the static website as `error.html`.

- Back-to-origin configurations

  In the following sample configuration, the <RoutingRules> field specifies back-to-origin rules. For more information about parameters that you can configure for back-to-origin rules, see Overview.

```
<?xml version="1.0" encoding="UTF-8"?>
 <WebsiteConfiguration>
     <IndexDocument>
         <Suffix>index.html</Suffix>
     </IndexDocument>
     <ErrorDocument>
         <Key>error.html</Key>
     </ErrorDocument>
     <RoutingRules>
         <RoutingRule>
             <RuleNumber>1</RuleNumber>
             <Condition>
                 <KeyPrefixEquals>abc/</KeyPrefixEquals>
                 <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
             </Condition>
             <Redirect>
                 <RedirectType>Mirror</RedirectType>
                 <PassQueryString>true</PassQueryString>
                 <MirrorURL>http://example.com/</MirrorURL>
                 <MirrorPassQueryString>true</MirrorPassQueryString>
                 <MirrorFollowRedirect>true</MirrorFollowRedirect>
                 <MirrorCheckMd5>false</MirrorCheckMd5>
                 <MirrorHeaders>
                   <PassAll>true</PassAll>
                   <Pass>myheader-key1</Pass>
                   <Pass>myheader-key2</Pass>
                   <Remove>myheader-key3</Remove>
                   <Remove>myheader-key4</Remove>
                   <Set>
                     <Key>myheader-key5</Key>
                     <Value>myheader-value5</Value>
                   </Set>
                 </MirrorHeaders>
             </Redirect>
         </RoutingRule>
         <RoutingRule>
             <RuleNumber>2</RuleNumber>
             <Condition>
               <KeyPrefixEquals>abc/</KeyPrefixEquals>
               <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
               <IncludeHeader>
                 <Key>host</Key>
                 <Equals>test.oss-cn-beijing-internal.aliyuncs.com</Equals>
               </IncludeHeader>
             </Condition>
             <Redirect>
               <RedirectType>AliCDN</RedirectType>
               <Protocol>http</Protocol>
               <HostName>example.com</HostName>
```

```
                <PassQueryString>false</PassQueryString>
                <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith>
                <HttpRedirectCode>301</HttpRedirectCode>
            </Redirect>
        </RoutingRule>
    </RoutingRules>
</WebsiteConfiguration>
```

ii. You can run the following command to add website-related configurations for the bucket named examplebucket:

```
./ossutil64 website --method put oss://examplebucket localfile.txt
```

If a similar output is displayed, the website-related configurations are added to examplebucket:

```
0.299514(s) elapsed
```

Query website-related configurations

- Command syntax

```
./ossutil64 website --method get oss://bucketname [local_xml_file]
```

The following table describes the parameters that you can configure when you run this command to query website-related configurations.

Parameter	Description
bucketname	The name of the bucket whose website-related configurations you want to query.
local_xml_file	The name of the local file used to store the website-related configurations. Example: `localfile.txt`. If this parameter is not specified, obtained website-related configurations are displayed without being stored in the local file.

- Examples
  - You can run the following command to query the website-related configurations for the examplebucket bucket and write the configurations into the `localfile.txt` local file:

    ```
    ./ossutil64 website --method get oss://examplebucket localfile.txt
    ```

    If a similar output is displayed, the website-related configurations for examplebucket are obtained and written into localfile.txt:

    ```
    0.212407(s) elapsed
    ```

  - You can run the following command to query the website-related configurations of the examplebucket bucket and specify that the output is displayed without being stored in the local file:

    ```
    ./ossutil64 website --method get oss://examplebucket
    ```

    If a similar output is displayed, the website-related configurations are obtained:

```
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
    <SupportSubDir>false</SupportSubDir>
    <Type>0</Type>
  </IndexDocument>
  <ErrorDocument>
    <Key>error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <RuleNumber>1</RuleNumber>
      <Condition>
        <KeyPrefixEquals>abc/</KeyPrefixEquals>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <RedirectType>Mirror</RedirectType>
        <PassQueryString>true</PassQueryString>
        <MirrorURL>http://example.com/</MirrorURL>
        <MirrorPassQueryString>true</MirrorPassQueryString>
        <MirrorPassOriginalSlashes>false</MirrorPassOriginalSlashes>
        <MirrorFollowRedirect>true</MirrorFollowRedirect>
        <MirrorIsExpressTunnel>false</MirrorIsExpressTunnel>
        <MirrorUserLastModified>false</MirrorUserLastModified>
        <MirrorCheckMd5>false</MirrorCheckMd5>
        <MirrorSwitchAllErrors>false</MirrorSwitchAllErrors>
        <MirrorHeaders>
          <PassAll>true</PassAll>
          <Pass>myheader-key1</Pass>
          <Pass>myheader-key2</Pass>
          <Remove>myheader-key3</Remove>
          <Remove>myheader-key4</Remove>
          <Set>
            <Key>myheader-key5</Key>
            <Value>myheader-value5</Value>
          </Set>
        </MirrorHeaders>
        <MirrorUsingRole>false</MirrorUsingRole>
        <MirrorAllowHeadObject>false</MirrorAllowHeadObject>
        <EnableReplacePrefix>false</EnableReplacePrefix>
      </Redirect>
    </RoutingRule>
    <RoutingRule>
      <RuleNumber>2</RuleNumber>
      <Condition>
        <IncludeHeader>
          <Key>host</Key>
          <Equals>test.oss-cn-beijing-internal.aliyuncs.com</Equals>
        </IncludeHeader>
        <KeyPrefixEquals>abc/</KeyPrefixEquals>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
```

```
      <Redirect>
        <RedirectType>AliCDN</RedirectType>
        <Protocol>http</Protocol>
        <HostName>example.com</HostName>
        <PassQueryString>false</PassQueryString>
        <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith>
        <EnableReplacePrefix>false</EnableReplacePrefix>
        <HttpRedirectCode>301</HttpRedirectCode>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
0.157648(s) elapsed
```

Delete website-related configurations

- Command syntax

  ```
  ./ossutil64 website --method delete oss://bucketname
  ```

- Examples

  You can run the following command to delete the website-related configurations for the bucket named examplebucket:

  ```
  ./ossutil64 website --method delete oss://examplebucket
  ```

  If a similar output is displayed, the website-related configurations for the bucket are deleted:

  ```
  0.212409(s) elapsed
  ```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the *-e* option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the *-i* option to use the AccessKey ID of the specified account, and use the *-k* option to use the AccessKey secret of the specified account.

For example, you can run the following command to add website-related configurations for a bucket named testbucket, which is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 website --method put oss://testbucket localfile.txt -e oss-cn-hangzhou.aliyuncs
.com -i LTAI4Fw2NbDUCV8zYUzA****  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that you can use for the website command, see Common options.

2.3.41. worm (manage retention policies)

OSS supports time-based retention policies. After a retention policy is configured for a bucket, objects stored inside the bucket cannot be deleted or modified during the retention period. This topic describes how to use the **worm** command to manage the retention policies configured for a bucket.

> **Note**
>
> - The commands described in this topic are for 64-bit Linux systems. To use the commands in the examples on 64-bit Windows systems, replace **./ossutil64** in the commands with **ossutil64.exe**.
> - For more information about retention policies, see Retention policy.

Create and lock a retention policy

To use a retention policy to protect objects in your bucket, you must create and lock the policy.

1. Create a retention policy.
   - Command syntax

     ```
     ./ossutil64 worm init oss://BucketName days
     ```

     The following table describes the parameters that you can configure when you use the worm command to create a retention policy.

Parameter	Description
BucketName	The name of the bucket for which you want to configure the retention policy.
days	The retention period of the retention policy. During the retention period, objects in the bucket cannot be modified or deleted. ■ Unit: days ■ Valid values: 1 to 25550

   - Examples

     You can run the following command to create a retention policy for a bucket named examplebucket and sets the retention period of the policy to 180 days:

     ```
     ./ossutil64 worm init oss://examplebucket 180
     ```

     If a similar output is displayed, the retention policy is created.

     ```
     init success,worm id is 581D8A7FFA064C80827CAB4076A93A78
     ```

2. Lock a retention policy.
   - Command syntax

     ```
     ./ossutil64 worm complete oss://BucketName WormId
     ```

     The following table describes the parameters that you can configure when you use the worm command to lock a retention policy.

Parameter	Description
BucketName	The name of the bucket for which the retention policy you want to lock is configured.

Parameter	Description
WormId	The ID of the retention policy you want to lock. This parameter is returned when you successfully create a retention policy.

- Examples

  You can run the following command to lock the retention policy configured for a bucket named examplebucket:

  ```
  ./ossutil64 worm complete oss://examplebucket 581D8A7FFA064C80827CAB4076A93A78
  ```

  If a similar output is displayed, the retention policy is locked.

  ```
  0.073810(s) elapsed
  ```

Extend the retention period of a retention policy

After a retention policy is locked, objects in the bucket cannot be modified or deleted during the retention period of the policy. If the retention period of the retention policy cannot meet your requirements for data protection, you can run the following command to extend the retention policy.

- Command syntax

  ```
  ./ossutil64 worm extend oss://BucketName days WormId
  ```

- Examples

  You can run the following command to extend the retention period of the retention policy configured for a bucket named examplebucket to 360 days:

  ```
  ./ossutil64 worm extend oss://examplebucket 360 581D8A7FFA064C80827CAB4076A93A78
  ```

  If a similar output is displayed, the retention period of the retention policy is extended to 360 days.

  ```
  0.067810(s) elapsed
  ```

Query the configurations of a retention policy

You can run the following command to query the configurations of the retention policy configured for a bucket.

- Command syntax

  ```
  ./ossutil64 worm get oss://BucketName
  ```

- Examples

  You can run the following command to query the configurations of the retention policy configured for a bucket named examplebucket:

  ```
  ./ossutil64 worm get oss://examplebucket
  ```

  If a similar output is displayed, the configurations of the retention policy are found. The returned results include the ID, status, retention period, and creation time of the retention policy.

```
<WormConfiguration>
      <WormId>581D8A7FFA064C80827CAB4076A93A78</WormId>
      <State>Locked</State>
      <RetentionPeriodInDays>360</RetentionPeriodInDays>
      <CreationDate>2021-01-19T03:36:53.000Z</CreationDate>
  </WormConfiguration>
```

Delete a retention policy

You can delete a retention policy before the policy is locked.

- Command syntax

```
./ossutil64 worm abort oss://BucketName
```

- Examples

  You can run the following command to delete the retention policy configured for a bucket named examplebucket:

```
./ossutil64 worm abort oss://examplebucket
```

  If a similar output is displayed, the retention policy is deleted.

```
0.067810(s) elapsed
```

Common options

To use ossutil to manage buckets that are located in different regions, you can use the -e option to use the endpoint of the specified bucket. To use ossutil to manage buckets that are owned by different Alibaba Cloud accounts, you can use the -i option to use the AccessKey ID of the specified account, and use the -k option to use the AccessKey secret of the specified account.

For example, you can run the following command to create a retention policy for a bucket named test, which is located in the China (Hangzhou) region and is owned by another Alibaba Cloud account:

```
./ossutil64 worm init oss://test -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA***
*  -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

For more information about other common options that apply to the worm command, see Common options.

2.4. View options

This topic describes how to run the -h command to view all options supported by ossutil.

> **Note** Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

Command syntax

```
./ossutil64 -h
```

To view the options supported by a command, run the **./ossutil64 help [command]** command, such as **./ossutil64 help cp**.

Common options

The following table describes common options that can be added to most commands supported by ossutil.

Option	Description
-c, --config-file	Specifies the path of the configuration file of ossutil. ossutil reads the configuration file when this command-line tool is started. When you manage buckets that belong to different Alibaba Cloud accounts, you can generate multiple configuration files, and specify one of these configuration files as the default configuration file. When you manage a bucket that belongs to another Alibaba Cloud account, you can use the *-c* option to specify the corresponding configuration file.
-e, --endpoint	Specifies the endpoints of buckets. When you manage buckets across regions, you can use this option to specify the corresponding endpoints of the buckets. For more information about endpoints of different regions, see Regions and endpoints.
-i, --access-key-id	Specifies the AccessKey ID used to access Object Storage Service (OSS). When you manage buckets that belong to different Alibaba Cloud accounts, you can use this option to specify the corresponding AccessKey IDs.
-k, --access-key-secret	Specifies the AccessKey secret used to access OSS. When you manage buckets that belong to different Alibaba Cloud accounts, you can use this option to specify the corresponding AccessKey secrets.
-p, --password	Specifies the AccessKey secret used to access OSS. When you use this option in a command, ossutil reads the AccessKey secret that is entered by using the keyboard and ignores the AccessKey secret configured by using other methods.

--loglevel	Generates the *ossutil.log* file in the current working directory. The default value is empty, which indicates that no log files are generated.
	Valid values:
	• *info*: ossutil displays operations logs.
	`./ossutil64 [command] --loglevel=info`
	• *debug*: ossutil displays logs that contain HTTP requests and responses and original signature strings to locate problems.
	`./ossutil64 [command] --loglevel=debug`

Option	Description
--proxy-host, --proxy-user, and --proxy-pwd	If your environment requires a proxy server to access websites, you must use these three options to specify the information of the proxy server. • --proxy-host: indicates the URL of the proxy server. This option supports *HTTP*, *HTTPS*, and *SOCKS5*. • --proxy-user: indicates the username of the proxy server. The default value is empty. • --proxy-pwd: indicates the password of the proxy server. The default value is empty. After you specify the information of the proxy server by using these three options, ossutil uses the specified information and the proxy server to access OSS `./ossutil64 ls oss://bucket1 --proxy-host http://47.88.**.**:3128 --proxy-user test --proxy-pwd test`
--mode	Specifies the type of your access credential. Valid values: • *AK*: The AccessKey ID and AccessKey secret are used for authentication. • *StsToken*: An Security Token Service (STS) token is used for authentication. • *RamRoleArn*: The AssumeRole method of Resource Access Management (RAM) is used for authentication. • *EcsRamRole*: The EcsRamRole method in an Elastic Compute Service (ECS) instance is used to implement password-free authentication. If you do not specify this option, the default authentication logic is used.
--ecs-role-name	Specifies the role name in EcsRamRole mode for authentication.
--token-timeout	Specifies the validity period of the temporary access credential token specified in the AssumeRole parameter in RamRoleArn mode for authentication. Default value: 3600. Unit: seconds.
--ram-role-arn	Specifies the Alibaba Cloud Resource Name (ARN) of the RAM role in RamRoleArn mode for authentication.
--role-session-name	Specifies the session name in RamRoleArn mode for authentication.
--read-timeout	Specifies the timeout period for the client to read data. Default value: 1200. Unit: seconds.
--connect-timeout	Specifies the timeout period for the client to connect to the server. Default value: 120. Unit: seconds.

Option	Description
--sts-region	Specifies the region where STS is connected. Format: `cn-hangzhou`. For more information about the regions supported by STS, see Endpoints. If you do not specify this option, the default value of this option in RamRoleArn mode for authentication is `sts.aliyuncs.com`.
--skip-verify-cert	Specifies that the digital certificate of the server is not verified.

Other options

The following table describes other options included in ossutil commands other than the preceding common options.

Option	Description
-s, --short-format	Lists items in the short format. The long format is displayed if this option is not specified.
--bigfile-threshold	Specifies the size threshold over which a large object starts resumable data transfer. Default value: 104857600 (100 MB). Valid values: 0 to 9223372036854775807. Unit: bytes.
--acl	Sets the access control list (ACL) for the object.
--range	Specifies the byte range of the object to download. Bytes are numbered from 0. • You can specify a range. For example, *3-9* indicates a range from byte 3 to byte 9, which includes byte 3 and byte 9. • You can specify the field from which the download starts. For example, *3-* indicates a range from byte 3 to the end of the object, which includes byte 3. • You can specify the field at which the download ends. For example, *-9* indicates a range from byte 0 to byte 9, which includes byte 9.
--all-versions	Specifies all versions of an object.
--type	Specifies the algorithm that is used for verification. Default value: crc64. Valid values: • *crc64* • *md5*: specifies the MD5 verification.
-v, --version	Displays the ossutil version and exits.
-u, --update	Specifies an update operation.
--origin	Specifies the value of the Origin header in an HTTP request.

Option	Description
--upmode	**probe**: specifies the upload method in the command. Default value: normal. Valid values: • *normal*: The object is uploaded by using simple upload. • *append*: The object is uploaded by using append upload. • *multipart*: The object is uploaded by using multipart upload.
--sse-algorithm	Specifies the encryption method for the bucket. Valid values: • *KMS*: The keys managed by Key Management Service (KMS) are used for encryption and decryption (SSE-KMS). • *AES256*: The keys managed by OSS are used for encryption and decryption (SSE-OSS).
--include	Specifies that objects that meet specified conditions are listed. For example, a value of `*.jpg` indicates that all objects in the JPG format are listed.
--exclude	Specifies that objects that do not meet specified conditions are listed. For example, a value of `*.txt` indicates that all objects that are not in the TXT format are listed.
-r, --recursive	Specifies recursive operations. If this option is specified, commands that support this option are run to perform operations on all objects in a bucket that meet specified conditions. If this option is not specified, commands that support this option are run to perform operations only on the specified object.
--addr	Specifies the address of the network that you want to check. ossutil runs the ping command to check the network connectivity between your local computer and the specified address. Default value: `www.aliyun.com`.
--kms-masterkey-id	Specifies the customer master key (CMK) ID used for encryption in KMS.
--version-id	Specifies the version ID of an object.
--version-id-marke	Specifies the position from which a list operation starts. Object versions whose IDs are alphabetically after the value of marker are listed. You can specify this parameter only after versioning is enabled for the bucket.
-m, --multipart	Specifies that operations are to be performed on the incomplete multipart upload tasks in a bucket.
-d, --directory	Specifies that only objects and subdirectories in the current directory are returned.
--payer	Specifies the payer of the request. If you set this option to *requester*, the pay-by-requester mode is enabled.
--maxupspeed	Specifies the maximum upload speed. Default value: 0 (unlimited). Unit: KB/s.

Option	Description
--maxdownspeed	Specifies the maximum download speed. Default value: 0 (unlimited). Unit: KB/s.
--retry-times	Specifies the number of retries after errors occur. Valid value: 1 to 500. Default value: 10.
--download	Specifies that ossutil checks the network between your local computer and a bucket by using the URL of an object to download the object from the bucket.
-j, --jobs	Specifies the number of concurrent tasks performed across multiple objects. Valid values: 1 to 10000. Default value: 3.
-a, --all-type	Specifies that operations are to be performed on the objects and incomplete multipart upload tasks in a bucket.
--disable-empty-referer	Specifies that the referer field is not allowed to be empty.
--method	Specifies the HTTP request method, which can be PUT, GET, or DELETE.
--output-dir	The directory in which the output objects are located. Output objects include report objects generated due to errors that occur when you run the **cp** command to copy multiple objects. Default value: the *ossutil_output* directory in the current directory.
--meta	Specifies the metadata of an object in the [header:value#header:value...] format. Example: `Cache-Control: no-cache#Content-Encoding: gzip`.
--object	Specifies an object name in **probe** commands.
--end-time	Specifies the timestamp in Linux or UNIX. If you specify this option, objects whose last modified time is later than the timestamp are ignored.
--limited-num	Specifies the maximum number of returned results.
-L, --language	Specifies the language that ossutil uses. Default value: CH. Valid values: • *CH*: Chinese. If you plan to set this option to CH, make sure that your system supports UTF-8 encoding. • *EN*: English.
--delete	Specifies operations to delete buckets, objects, or parts.
-b, --bucket	Specifies that the command is run to delete a bucket. This option can be specified only when you run the rm command.
--disable-crc64	Disables CRC-64 in data transmission. By default, CRC-64 is enabled when you use ossutil to transfer data.

Option	Description
--upload	Specifies that the connection between your local computer and the destination bucket is checked by uploading a local file to the destination bucket.
--part-size	Specifies the part size in bytes. By default, ossutil calculates the appropriate part size based on the object size. You can set this option to optimize performance or meet special requirements. Valid values: 1 to 9223372036854775807.
--timeout	Specifies the timeout period of a signed URL. Valid values: 0 to 9223372036854775807. Default value: 60. Unit: seconds.
--checkpoint-dir	Specifies the directory in which the log objects of resumable upload or download tasks are stored. Default value: `.ossutil_checkpoint`. When a resumable synchronization task fails, ossutil creates this directory and stores the checkpoint information about the task in this directory. ossutil deletes the directory after the task is complete. If this option is specified, make sure that you have permissions to delete the specified directory.
--url	Specifies the URL of an object.
--marker	Specifies the position from which the list operation starts. Buckets, objects, and parts whose names are alphabetically after the value of marker are listed.
-f, --force	Forces an operation without prompting the user for confirmation.

Option	Description
--snapshot-path	Specifies the directory in which the snapshots of uploaded objects and downloaded objects are stored. If this option is specified when objects are uploaded or downloaded, ossutil reads the snapshot information from the specified directory and performs an incremental upload or download. • The --snapshot-path option is used to accelerate the incremental upload or download of multiple objects. This option cannot be used to copy objects. This option can be used when the number of objects is large and no other users modify the corresponding objects in OSS between the two uploads. • The --snapshot-path option records the last modified time of uploaded or downloaded objects in your local computer. Then, the recorded time is used to determine which objects can be skipped next time the objects are uploaded or downloaded. Therefore, when you use this option, make sure that the corresponding objects in OSS are not modified between the two uploads or downloads. In other scenarios where objects are updated in OSS between the two uploads or downloads, use the --update option to perform incremental upload or download on objects. • ossutil does not automatically delete snapshot information from the directory specified by snapshot-path. You can delete snapshot information that you no longer use. • Additional overheads are required to read and write snapshot information. We recommend that you do not use this option in the following scenarios: The number of objects to upload or download is small. Network conditions are good. Other users need to perform operations on those objects. In this case, you can use the --update option to perform incremental upload or download. • You can specify the --update and --snapshot-path parameters at the same time in a command. ossutil determines whether to skip an object in uploads or downloads first based on the snapshots stored in the directory specified by --snapshot-path. If no snapshots are generated for the object, ossutil determines whether to skip the object based on the --update option.
--start-time	Specifies the timestamp in Linux or UNIX. If you specify this option, objects whose last modified time is earlier than the timestamp are ignored.

Option	Description
--storage-class	Specifies the storage class of an object. Default value: Standard. Valid values: • *Standard*: This storage class is suitable for objects that are frequently accessed. • *IA*: This storage class is suitable for objects that are infrequently accessed (once or twice each month). Objects of the Infrequent Access (IA) storage class have a minimum storage period of 30 days and a minimum billable size of 64 KB. You can access objects of the IA storage class in real time. You are charged data retrieval fees when you access IA objects. • *Archive*: This storage class is suitable for objects that are stored for a long period of time. Objects of the Archive storage class have a minimum storage period of 60 days and a minimum billable size of 64 KB. You must restore an Archive object before you can access it. The restoration takes about one minute, and you are charged data retrieval fees. • *ColdArchive*: This storage class is suitable for objects that are barely accessed. Objects of the Cold Archive storage class have a minimum storage period of 180 days and a minimum billable size of 64 KB. You must restore an object of the Cold Archive storage class before you can access the object. The time required to restore a Cold Archive object depends on the object size and the restore mode. You are charged the data retrieval fees when you restore a Cold Archive object.
-t, --sts-token	Specifies the STS token used to access OSS. This option is required only when you use a temporary STS token to access OSS. This option value overwrites the corresponding configurations in the configuration file. For more information about how to generate an STS token, see Authorized third-party upload.
--parallel	Specifies the number of concurrent operations performed on a single object. Valid values: 1 to 10000. By default, ossutil automatically sets the value of this option based on the operation type and object size.
--partition-download	Specifies the partition in which the object that you want to download is stored. The value of this option is in the `partition number: the total number of partitions` format. For example, a value of 1:5 indicates that ossutil downloads partition 1 out of the five partitions. Partitions are numbered from 1. Partitioning rules for objects are determined by ossutil. This option splits an object into multiple partitions that can be concurrently downloaded by multiple ossutil commands. Each ossutil command downloads its own partition.
--bucketname	Specifies the name of a bucket.
--encoding-type	Specifies the encoding type used to encode the key that follows `oss://bucket_name`. Valid value: *url*. If this option is not specified, the key is not encoded.
--origin	Specifies the value of the Origin header in an HTTP request. This option value indicates the source domain of a cross-origin request.

Option	Description
--acr-method	Specifies the value of the Access-Control-Request-Method header in an HTTP request. Valid values: GET, PUT, POST, DELETE, and HEAD.
--acr-headers	Specifies the value of the Access-Control-Request-Headers request header. The value of this option indicates the headers that you want to add to the request. The headers do not include common request headers. To specify multiple headers, separate different headers with commas (,) and enclose the headers with double quotation marks ("). Example: `--acr-headers "header1,header2,header3"`.
--upload-id-marker	Specifies the position from which the list operation starts. Parts whose upload IDs are alphabetically after the value of marker are listed.
-h, --help	Displays help information for a specified command.
--trafic-limit	Specifies the access speed over HTTP in **sign** commands. Unit: bit/s. Default value: 0. A value of 0 indicates that the access speed is unlimited. Valid values: 819200 to 838860800 (100 KB/s to 100 MB/s).
--local-host	Specifies the local IP address of ossutil in **cp** commands. After you specify this option, ossutil accesses OSS by using the specified IP address.
--enable-symlink-dir	Specifies that the subdirectory to which the symbolic link points are uploaded. By default, subdirectories are not uploaded. The probe command can be used to check whether an object or a directory to which the symbolic link points is also a symbolic link.
--only-current-dir	Specifies that only objects in the current directory are uploaded, downloaded, or copied. Subdirectories and objects in the subdirectories in the current directory are ignored.
--disable-dir-object	Specifies that no OSS object is generated for the directory to upload, but you can still view the corresponding directory structure in the OSS console. If you delete all objects in the directory, the directory is also deleted.
--probe-item	Specifies the items to be checked by using the probe command. Valid values: • *upload-speed*: ossutil checks the upload bandwidth. • *download-speed*: ossutil checks the download bandwidth. • *cycle-symlink*: ossutil checks whether a symbolic link in the local file directory points to itself.

Option	Description
--redundancy-type	Specifies the redundancy type of a bucket. Default value: LRS. Valid values: • *LRS*: If you set the redundancy type of a bucket to locally redundant storage (LRS), OSS stores the copies of each object across different devices within the same zone. This way, OSS ensures data reliability and availability when hardware failures occur. • *ZRS*: If you set the redundancy type of a bucket to zone-redundant storage (ZRS), OSS uses the multi-zone mechanism to store data across three zones within the same region. Even if one zone becomes unavailable due to failures such as power outages and fires, the data is still accessible.
--disable-encode-slash	Specifies that forward slashes (/) in a URL are not encoded.
--disable-all-symlink	Specifies that all objects in the subdirectory to which the symbolic link points and the subdirectory to which the symbolic link points are ignored during object upload.
--tagging	Specifies the object tag when you upload or copy an object in the `"abc=1&bcd=2&....."` format.
--disable-ignore-error	Specifies that errors are not ignored during batch operations.
--block-size	Specifies the unit of the obtained total size of objects within the specified bucket or directory. Valid values: KB, MB, GB, and TB. By default, if you do not specify this option, the obtained total size of objects is measured in bytes. **Notice** This option applies to ossutil version 1.7.3 and later.

2.5. FAQ

This topic describes the problems you may encounter when you use ossutil and their solutions.

Note Sample command lines in this topic are based on the 64-bit Linux system. For other systems, replace **./ossutil64** in the commands with the corresponding binary name. For more information, see ossutil.

What do I do if the skip message appears when the -u parameter is used to upload an object?

Analysis: When you use the -u parameter to upload an object, ossutil compares the object to upload with the existing objects in the bucket. When the name of the object to upload is the same as that of an existing object, if the last modified time of the object to upload is equal to or earlier than that of the existing object, the object is skipped. If the last modified time of the object to upload is later than that of the existing object, the object is uploaded. Therefore, the skip message is normal when you use the -u parameter to upload an object.

Solution: Ignore the message if the object is uploaded.

What do I do if a 403 error is returned when an object is restored?

Analysis: This error is returned because of the following reasons:

- The RAM user that you use to restore the object does not have corresponding permissions.
- Access to the object is blocked because because its content is illegal.

Solution:

- Grant permissions to the RAM user.
- Delete or ignore the object.

What do I do if an error is returned when a bucket that does not contain objects according to the result of the ls command is deleted?

Analysis: When you run the **ls** command without options included to list objects in a bucket, the parts and previous versions (only for versioned buckets) in the bucket are not listed. A bucket that contains parts or previous versions cannot be deleted by the **rm** command.

Solution:

- Delete parts and previous versions (only for versioned buckets) in a bucket before you delete the bucket.
  i. Delete parts and previous versions in a bucket.
     - Run the following commands to list and delete parts in the bucket:

       ```
       ./ossutil64 ls oss://bucket1 -m
       ./ossutil64 rm -m oss://bucket1 -r
       ```

     - Run the following commands to list and delete previous versions in the bucket:

       ```
       ./ossutil64 ls oss://bucket1 --all-versions
       ./ossutil64 rm oss://bucket1 --all-versions -r
       ```

  ii. Delete the bucket.

      ```
      ./ossutil64 rm oss://bucket1 -b
      ```

- Force delete the bucket.
  - Run the following command to force delete an unversioned bucket:

    ```
    ./ossutil64 rm oss://bucketname -abrf
    ```

  - Run the following command to force delete a versioned bucket:

    ```
    ./ossutil64 rm oss://bucketname -abrf --all-versions
    ```

Warning Deleted buckets and objects cannot be recovered. Exercise caution when you run the preceding commands.

What do I do if the upload or download progress exceeds 100%?

Analysis: A folder named *.ossutil_checkpoint* is generated when you use ossutil to upload or download objects. By default, if the object to upload or download is larger than 100 MB, ossutil uses resumable upload or download. The checkpoint file generated during the upload or download is stored in the *.ossutil_checkpoint* folder. This folder is deleted after the upload or download task is completed. In scenarios where multiple ossutil instances are running on a computer at the same time and each instance is used to perform resumable upload or download, when the upload or download task in an instance is completed, the *.ossutil_checkpoint* folder is deleted. In this case, the upload or download tasks in other instances cannot be completed and the progress exceeds 100%.

Solution:

- Complete the current upload or download tasks and upload or download the object again.
- Add the --checkpoint-dir parameter to the **cp** command and specify a folder whose name is different from the default checkpoint folder to store the checkpoint file. Example:

```
./ossutil64 cp oss://bucket1/myphoto.jpg  /dir  --checkpoint-dir checkpoint
```

What do I do if multiple requests are recorded in OSS logs when you send a request to download a single object by using ossutil?

Analysis: This error is returned because of the following reasons:

- ossutil retries the request because exceptions occur or the request fails. By default, ossutil retires a request for up to 10 times.
- If the object you want to download is larger than 100 MB in size, ossutil sends multiple requests to obtain data by range. Parts of the object data in a range are obtained by each request until the whole object is downloaded. Therefore, multiple request records are generated when an object larger than 100 MB in size is downloaded.

3.ossbrowser

3.1. Install and log on to ossbrowser

ossbrowser is a graphical management tool developed by Alibaba Cloud. This tool provides features similar to Windows Explorer. This topic describes how to install and log on to ossbrowser.

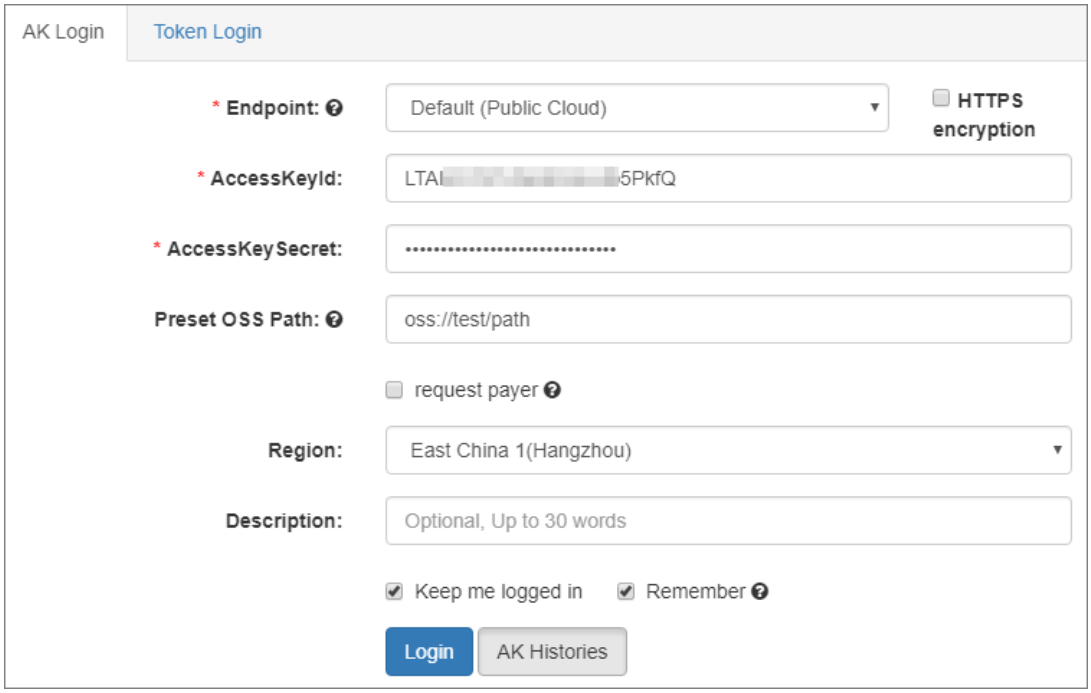
Procedure

1. Download and install ossbrowser.

Current version	Operating system	Description
1.16.0	Windows x32	Windows 7 and later are supported.
	Windows x64	
	Windows Server (not recommended)	If issues occur when you install ossbrowser on Windows Server, build source code on your own to resolve compatibility issues.
	macOS	Applications that are not verified by developers cannot be run on macOS. If this issue occurs, modify the security settings of your system. For more information, see What do I do if I cannot run ossbrowser on macOS?.
	Linux x64 (not recommended)	If issues occur when you install ossbrowser on Linux, download and compile the source code to troubleshoot the issues.

2. Log on to ossbrowser. In this topic, Windows is used in the example to describe the procedure.
   i. Double-click `oss-browser.exe` .
   ii. Use one of the following methods to log on to ossbrowser:
   - Log on to ossbrowser by using an AccessKey pair

     You can use the AccessKey pair of an Alibaba Cloud account or a RAM user to log on to ossbrowser.



The following table describes the parameters that you can configure when you log on to ossbrowser by using an AccessKey pair.

Parameter	Description
Endpoint	Select the endpoint of the region that you want to access. ■ **Default (Public Cloud)**: Use the endpoint of the region in which the current bucket is located to log on to ossbrowser. When you select this value, you can select **HTTPS encryption** to encrypt data transmission. ■ **Customize**: Use the endpoint of another region to log on to ossbrowser. Example: `https://oss-cn-beijing.aliyuncs.com`. For more information about regions and endpoints, see Regions and endpoints. ■ **cname**: Use a custom domain name to access Object Storage Service (OSS) resources. You must map a custom domain name to the bucket that you want to access in advance. For more information, see Map custom domain names. After you map a custom domain name to the bucket, enter the custom domain name to access the bucket.

Parameter	Description
AccessKeyId and **AccessKeySecret**	Enter the AccessKey pair of your account. For more information about how to obtain the AccessKey pair, see Obtain an AccessKey pair. **Notice** To ensure data security, we recommend that you log on to ossbrowser by using the AccessKey pair of a Resource Access Management (RAM) user. Before you use the AccessKey pair of a RAM user to log on to ossbrowser, you must grant the following permissions to the RAM user: `AliyunOSSFullAccess`, `AliyunRAMFullAccess`, and `AliyunSTSAssumeRoleAccess`. For more information about how to grant the permissions, see Permission management.
Preset OSS Path	If the current account has only permissions to access a specific bucket or a specific path in a bucket, you must specify this parameter in the following format: *oss://bucketname/path*. For example, if you are authorized to access only objects or subdirectories in the examplefolder directory of a bucket named examplebucket, enter *oss://examplebucket/examplefolder/*. If pay-by-requester is enabled for the bucket that you are authorized to access and you are not the bucket owner, select **request payer**. Otherwise, the `AccessDenied` error is returned when you access the resources specified by Preset OSS Path. If you select **request payer**, you are charged for the traffic and requests that are generated when you access the resources specified by Preset OSS Path. For more information about how to enable pay-by-requester, see Enable pay-by-requester.
Region	If you select **Default (Public Cloud)** for Endpoint, you must specify the region of the bucket in which the resources specified by Preset OSS Path are stored.
Keep me logged in	If you select this option, ossbrowser remains logged on next time you start ossbrowser.
Remember	If you select this option, your AccessKey pair used to log on to ossbrowser is saved. Next time you log on to ossbrowser, click **AK Histories** and select the saved AccessKey pair instead of entering the AccessKey pair again. **Warning** For security reasons, do not select this option if you use a shared computer.

- Log on to ossbrowser by using an authorization code

  To authorize other users to temporarily access specific resources in your buckets, you can generate an authorization code that can be used by these users to log on to ossbrowser. The authorization code becomes invalid after it expires. For more information about how to use an authorization code to log on to ossbrowser, see Log on to ossbrowser by using STS tokens.

3.2. Use ossbrowser

This topic describes how to use ossbrowser. ossbrowser is a graphical management tool developed by Alibaba Cloud. This tool provides features similar to Windows Explorer. You can use ossbrowser to manage your objects and buckets.

Prerequisites

ossbrowser is installed, and you already log on to ossbrowser. For more information, see Install and log on to ossbrowser.

Operations related to buckets and objects

ossbrowser supports the same operations related to buckets and objects as the Object Storage Service (OSS) console. You can follow the instructions of ossbrowser to manage your buckets and objects.

Operation	Description
Bucket-related operations	• Create a bucket A bucket is a container for objects stored in OSS. Before you upload an object to OSS, you must create a bucket. For more information about how to specify the name, region, access control list (ACL), and storage class of a bucket when you create the bucket, see Create buckets. • Delete a bucket If you no longer need a bucket, you can delete it to avoid incurring unnecessary charges. For more information, see Delete buckets.

Operation	Description
Object-related operations	ossbrowser supports operations such as upload, download, preview, move, and copy objects as well as generate object URLs. When you use ossbrowser to perform object-related operations, take note of the following items: • By default, ossbrowser uses multipart upload and resumable upload to upload objects. The object to upload cannot exceed 48.8 TB in size. If an upload is interrupted before the object is completely uploaded, the uploaded portion is stored as parts in an OSS bucket. To avoid unnecessary charges, we recommend that you use the following methods to delete these parts that you no longer need: ◦ Manually delete parts. For more information, see Manage parts. ◦ Configure lifecycle rules to automatically delete parts. For more information, see Configure lifecycle rules. Notice When the uploaded object has the same name as an existing object in the bucket, the existing object is overwritten. • The object that you can move or copy by using ossbrowser cannot exceed 5 GB in size. To move or copy objects that are larger than 5 GB, we recommend that you use ossutil. • After the object is uploaded to the bucket, you can share the object URL with third parties for downloads or previews. For more information, see Share objects. For more information about other object-related operations, see OSS Console User Guide.

3.3. Permission management

This topic describes how to perform simple permission management by using ossbrowser.

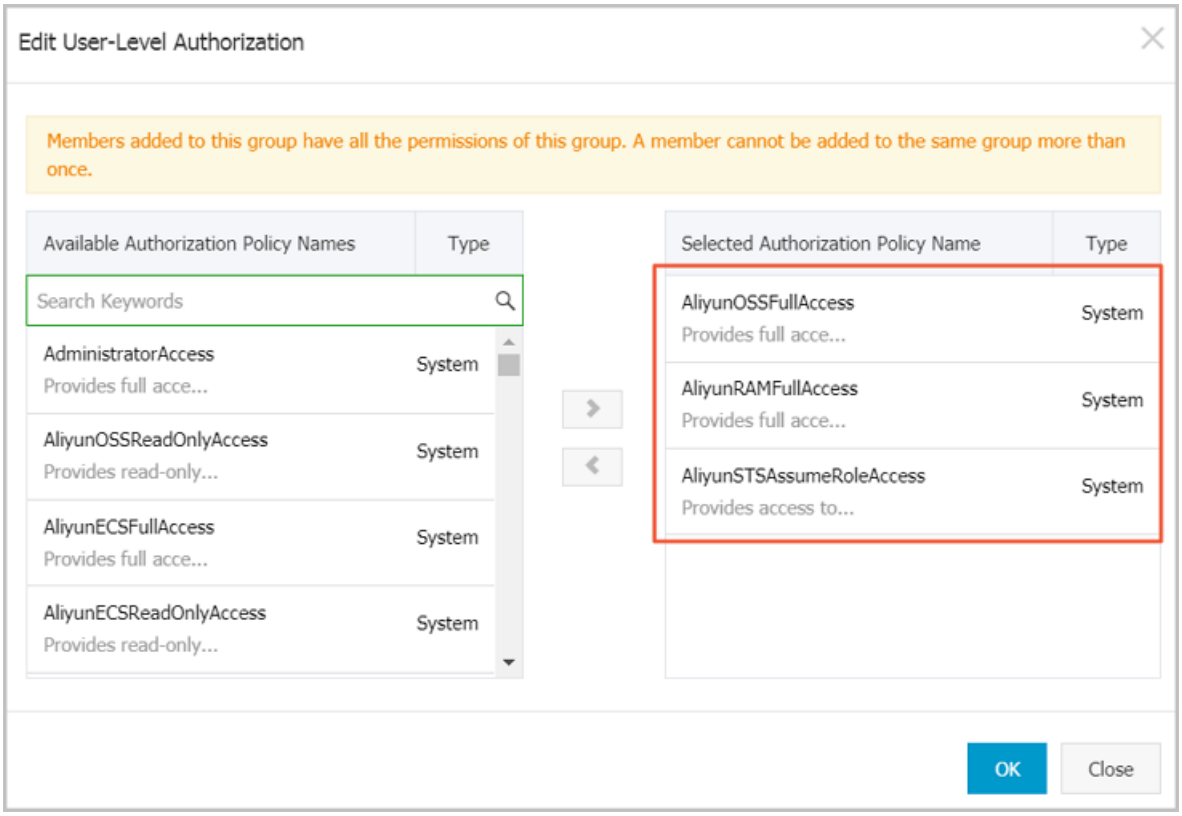
Log on to ossbrowser as a RAM user

For data security, we recommend that you use the AccessKey pair of a Resource Access Management (RAM) user to log on to ossbrowser.

Note For more information about how to create a RAM user and an AccessKey pair, see Create a RAM user.

RAM users can be classified into the following types based on their permissions:

- Administrator RAM user: a RAM user who has administrative permissions. For example, a RAM user who can manage all buckets and authorize other RAM users is an administrator RAM user. You can log on to the RAM console by using your Alibaba Cloud account to create an administrator RAM user and grant the permissions to the user. The following figure shows the permissions.



- Operator RAM user: a RAM user who has the read-only permission on a bucket or directory. Administrator RAM users can use the simple policy feature to grant operator RAM users permissions. For more information, see Grant permissions by using a simple policy.

> **Note** You can grant fine-grained permissions to RAM users. For more information, see Overview.

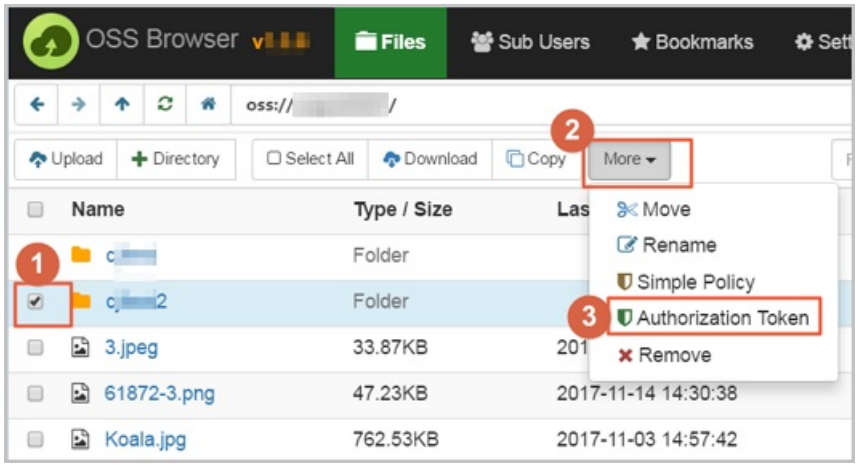
Log on to ossbrowser by using STS tokens

You can use an Security Token Service (STS) token to log on to ossbrowser. STS tokens can be provided for other authorized users for temporary access to a directory in your bucket. The STS token automatically becomes invalid after it expires.

1. Log on to ossbrowser as an administrator RAM user.

   > **Notice** For your data security, when you log on to ossbrowser by using your Alibaba Cloud account or as an administrator RAM user, part of the features are inaccessible. Use the AccessKey pair of an administrator RAM user to log on to ossbrowser and generate a token. The administrator RAM user must have the permissions to manage a bucket or directory, manage RAM (AliyunRAMFullAccess), and call the STS AssumeRole operation (AliyunSTSAssumeRoleAccess).

2. Select the objects or directories to be temporarily accessed by the authorized users, and choose **More > Authorization Token** shown in the following figure.



3. Save the obtained token.
4. Refer to the following figure, log off from ossbrowser, and then use the STS token to log on.

Grant permissions by using a simple policy

After you log on to ossbrowser as an administrator RAM user, you can use the **Simplify Policy** feature to create an operator RAM user, or grant an operator RAM user the read-only or read/write permissions on a bucket or directory.

> **Note** The simple policy feature of ossbrowser is designed based on Alibaba Cloud RAM to control access. You can log on to the RAM console from the Alibaba Cloud website to manage your RAM users more precisely.

1. Log on to ossbrowser as an administrator RAM user.
2. Select one or more objects or directories to be temporarily accessed by the authorized users, and choose **More > Simple Policy**.
3. In the **Simplify policy authorization** dialog box, set Privileges.
4. Grant permissions to an existing operator RAM user or create a new operator RAM user in this dialog box.

You can view, copy, and use the generated policy text. For example, you can copy the policy text and use it to edit the authorization policies for RAM users and roles in the RAM console.

> **Notice** To use the simple policy feature, you must log on to ossbrowser by using the AccessKey pair of a RAM user who has the RAM configuration permissions. For example, use the AccessKey pair of an administrator RAM user who has the RAM configuration permissions.

3.4. Troubleshooting

This topic provides answers to frequently asked questions about ossbrowser.

What do I do if the "AccessDenied:You are forbidden to list buckets" error appears when I use an AccessKey pair to log on to OSS from ossbrowser?

Cause: You do not have permissions to access all buckets.

Solution:

- Grant your Alibaba Cloud account the permissions and then log on to ossbrowser.
- If your Alibaba Cloud account is not authorized to access all of the objects or buckets, log on to ossbrowser by using the AccessKey pair of an Alibaba Cloud account that has permissions to access all objects or buckets. Set **Preset OSS Paths** and **Region**.

How do I increase the number of concurrent uploads and downloads when I need to upload or download a large number of objects?

Cause: By default, ossbrowser performs three upload or download tasks at the same time while the remaining tasks wait in the queue. Adjusting the number of concurrent uploads and downloads in the default setting will affect the transfer speed.

Solution: On the homepage of ossbrowser, click **Settings** in the top navigation bar. In the Settings dialog box, modify Transfer Settings. The number of concurrent uploads and downloads supported is determined by the hardware on which ossbrowser is installed. We recommend that you adjust and test the settings multiple times to optimize performance.

What do I do if I cannot run ossbrowser on macOS?

Cause: Applications that are not verified by developers cannot be run on macOS due to security settings.

Solution: Modify the security settings on macOS. Allow applications to run from any source.

1. On the command line in Linux, enter a command to disable the safe mode.

```
sudo spctl --master-disable
```

2. On the desktop of macOS, choose **Launchpad > System Preferences > Security & Privacy > General**.
3. Click the  icon in the lower-left corner. In the dialog box that appears, set User Name and Password. Click **Unlock**.
4. In the **Allow apps downloaded from** section, click **Anywhere**.

   After you complete these configurations, you can run ossbrowser on macOS.

What do I do if I cannot run ossbrowser after I install a software on Windows?

Cause: The software that you install adds the `NODE_OPTIONS=--max-http-header-size=value` environment variable to Windows.

Solution: Delete the `NODE_OPTIONS=--max-http-header-size=value` environment variable. The following example describes how to delete the environment variable on the 64-bit version of Windows 10:

1. Open the Control Panel. Click *System and Security*. On the page that appears, Click **System**.
2. In the left-side navigation pane, click **Advanced system settings**.
3. On the Advanced tab of the **System Properties** dialog box, click **Environment Variables**.
4. In the **User variables** list, select **NODE_OPTIONS=--max-http-header-size=*value***, and then click **Delete**.
5. Click **OK**.

   After you complete these configurations, you can run ossbrowser in Windows.

How do I enable the debugging mode?

You can use one of the following methods to enable the debugging mode:

- Method 1: Click the OSS Browser logo in the upper-left corner more than 10 consecutive times to go to the debugging panel.
- Method 2: In the Settings dialog box, click **Open debug** to go to the debugging panel.

References

For more information about frequently asked questions about ossbrowser, see FAQ.

4.ossimport

4.1. Architectures and configurations

ossimport is a tool used to migrate data to Object Storage Service (OSS). You can deploy ossimport on local servers or Elastic Compute Service (ECS) instances in the cloud to migrate data stored locally or in other cloud storage systems to OSS.

ossimport has the following features:

- Supports a wide range of data sources, including local data sources, Qiniu Cloud Object Storage (KODO), Baidu Object Storage (BOS), Amazon Simple Storage Service (Amazon S3), Azure Blob, UPYUN Storage Service (USS), Tencent Cloud Object Service (COS), Kingsoft Standard Storage Service (KS3), HTTP, and OSS. Additional sources can be added based on your requirements.
- Supports standalone and distributed modes. The standalone mode is easy to deploy and use. The distributed mode is suitable for large-scale data migration.
- Supports resumable data transfer.
- Supports throttling.
- Supports migration of objects whose last modified date is later than a specified time or objects whose names contain a specified prefix.
- Supports the upload and download of data in parallel.

Runtime environment

ossimport can be deployed on Linux or Windows systems that meets the following requirements:

- Windows 7 or later
- The latest version of Linux
- Java 1.7

> **Notice** ossimport cannot be deployed in distributed mode on Windows.

Deployment modes

ossimport supports the standalone and distributed modes.

- The standalone mode is sufficient for the migration of data smaller than 30 TB in size. You can deploy ossimport on a machine that can access the data to migrate and the OSS bucket to which you want to migrate the data.
- The distributed mode is suitable for the migration of data larger than 30 TB in size. You can deploy ossimport on multiple machines that can access the data to migrate and the OSS bucket to which you want to migrate the data.

> **Note** To reduce the time it takes to migrate large amounts of data, you can deploy ossimport on an ECS instance in the same region as your OSS bucket. Then, you can use a leased line to attach the server that stores the data to migrate to Alibaba Cloud Virtual Private Cloud (VPC). Transfer speed is greatly improved when the internal network is used to migrate data from ECS instances to OSS.

Standalone mode

Master, Worker, Tracker, and Console are packaged as `ossimport2.jar` and run on a machine. The system has only one Worker.

The following code describes the file structure in standalone mode:

```
ossimport
├── bin
│ └── ossimport2.jar  # The JAR package that contains the Master, Worker, Tracker, and Cons
ole modules.
├── conf
│ ├── local_job.cfg   # The Job configuration file for the standalone mode.
│ └── sys.properties  # The configuration file that contains system parameters.
├── console.bat         # The command-line tool in Windows used to run tasks step by step.
├── console.sh          # The command-line tool in Linux used to run tasks step by step.
├── import.bat          # The script that automatically imports files based on the conf/loc
al_job.cfg configuration file in Windows. The configuration file contains parameters that s
pecify data migration operations such as start, migration, verification, and retry.
├── import.sh           # The script that automatically imports files based on the conf/loc
al_job.cfg configuration file in Linux. The configuration file contains parameters that spe
cify data migration operations such as start, migration, verification, and retry.
├── logs                # The directory that contains logs.
└── README.md           # The file that introduces or explains ossimport. We recommend that
you read this file before you use ossimport.
```

- import.bat and import.sh are scripts that automatically import files based on the configuration file. You can run these tools after you modify the `local_job.cfg` configuration file.
- console.bat and console.sh are command-line tools used to run commands step by step.
- Run scripts or commands in the `ossimport` directory. These scripts and the `*.bat/*.sh` file are at the same directory level.

Distributed mode

The ossimport architecture in distributed mode consists of Master and Worker. The following code describes the structure:

```
Master --------- Job --------- Console
    |
    |
   TaskTracker
    |_____________________
    |Task     | Task      | Task
    |         |           |
Worker      Worker      Worker
```

Parameter	Description

Parameter	Description
Master	Splits a job into multiple tasks by data size and number of files. The data size and number of files can be configured in the sys.properties file. The master splits a job into multiple tasks by performing the following steps: 1. The master traverses the full list of files to migrate from the local device or the cloud storage system. 2. The master splits a job into multiple tasks by data size and number of files. Each task is responsible for the migration or verification of a portion of files.
Worker	• Migrates files and verifies data for tasks. A Worker pulls the specific file from the data source and uploads the file to the specified directory in OSS. You can specify the data source to migrate and OSS configurations in the job.cfg or local_job.cfg configuration file. • Supports throttling and specifies the number of concurrent tasks for data migration. You can configure the settings in the sys.properties configuration file.
TaskTracker	Distributes tasks and tracks task statuses. It is abbreviated to Tracker.
Console	Interacts with users and receives and displays command output. The console supports system management commands such as deploy, start, and stop, and job management commands such as submit, retry, and clean.
Job	Indicates the data migration jobs submitted by users. One job corresponds to one configuration file `job.cfg` .
Task	Migrates a portion of files. A job can be divided into multiple tasks by data size and number of files. The minimal unit for dividing a job into tasks is a file. One file is not assigned to multiple tasks.

In distributed mode, multiple workers can be started to migrate data. Tasks are evenly allocated to Workers. One worker can run multiple tasks. Only one worker can be started on each machine. The master and the tracker are started on the machine where the first worker specified by `workers` resides. The console must also run on this machine.

The following code describes the file structure in distributed mode:

```
ossimport
├── bin
│   ├── console.jar     # The JAR package for the Console module.
│   ├── master.jar      # The JAR package for the Master module.
│   ├── tracker.jar     # The JAR package for the Tracker module.
│   └── worker.jar      # The JAR package for the Worker module.
├── conf
│   ├── job.cfg         # The Job configuration file template.
│   ├── sys.properties  # The configuration file that contains system parameters.
│   └── workers         # The list of Workers.
├── console.sh          # The command-line tool. Currently, only Linux is supported.
├── logs                # The directory that contains logs.
└── README.md           # The file that introduces or explains ossimport. We recommend that
you read this file before you use ossimport.
```

Configuration file

The standalone mode has two configuration files *sys.properties* and *local_job.cfg*. The distributed mode has three configuration files *sys.properties*, *job.cfg*, and *workers*. The *local_job.cfg* and *job.cfg* configuration files have the same parameters. The *workers* configuration file is exclusive to the distributed mode.

- sys.properties: the system parameters

Parameter	Meaning	Description
workingDir	The working directory	The directory to which the tool package is decompressed. Do not modify this parameter in standalone mode. Working directories of each machine in distributed mode must be the same.
workerUser	The SSH username used to log on to the machine where Worker resides	◦ If privateKeyFile is configured, the value specified for privateKeyFile is used. ◦ If privateKeyFile is not configured, the values specified for workerUser and workerPassword are used. ◦ Do not modify this parameter in standalone mode.
workerPassword	The SSH password used to log on to the machine where Worker resides	Do not modify this parameter in standalone mode.

Parameter	Meaning	Description
privateKeyFile	The path of the private key file	◦ If you establish an SSH connection, you can specify this parameter. Otherwise, leave this parameter empty. ◦ If privateKeyFile is configured, the value specified for privateKeyFile is used. ◦ If privateKeyFile is not configured, the values specified for workerUser and workerPassword are used. ◦ Do not modify this parameter in standalone mode.
sshPort	The SSH port	The default value is 22. We recommend that you retain the default value. Do not modify this parameter in standalone mode.
workerTaskThreadNum	The maximum number of threads for Worker to run tasks	◦ This parameter is related to the machine memory and network conditions. We recommend that you set this parameter to 60. ◦ The value can be increased. For example, you can set this parameter to a greater value such as 150 for physical machines. If the maximum network bandwidth is reached, do not further increase the value. ◦ If the network conditions are poor, lower the value to such as 30. This way, you can avoid request timed out errors from limited bandwidth.
workerMaxThroughput(KB/s)	The traffic throttling of data migration for Worker	This value can be used for throttling. The default value is 0, which indicates that no throttling is imposed.
dispatcherThreadNum	The number of threads for task distribution and status confirmation of Tracker	If you do not have special requirements, retain the default value.
workerAbortWhenUncatchedException	Indicates whether to skip or stop a task if an unknown error occurs.	By default, unknown errors are skipped.

Parameter	Meaning	Description
workerRecordMd5	Indicates whether to use metadata x-oss-meta-md5 to record the MD5 hash values of files to migrate. By default, MD5 hash values are not recorded.	This parameter value is used to verify data integrity of files to migrate.

- job.cfg: the configurations for data migration jobs. The configuration files `local_job.cfg` and `job.cfg` have the same parameters.

Parameter	Meaning	Description
jobName	The name of the job. The value is of the String type.	◦ The unique identifier of the job. A job name has the following naming conventions: The name can contain letters, digits, underscores (_), and hyphens (-). The name must be 4 to 128 characters in length. You can submit multiple jobs with different names. ◦ If you submit a job with the same name as an existing job, the system prompts that the job already exists. Before you clean the existing job, you are not allowed to submit the job with the same name
jobType	The type of the job. The value is of the String type.	Valid values: import and audit. Default value: import. ◦ import: runs the data migration job and verifies the migration data for consistency. ◦ audit: only verifies data consistency.
isIncremental	Indicates whether to enable the incremental migration mode. The value is of the Boolean type.	◦ Default value: false ◦ If this parameter is set to true, incremental data is rescanned at the interval specified by incrementalModeInterval in seconds and is synchronized to OSS.
incrementalModeInterval	The synchronization interval in seconds in incremental mode. The value is of the Integer type.	This parameter is valid when isIncremental is set to true. The minimum configurable interval is 900 seconds. We recommend that you do not set it to a value smaller than 3600 seconds. If you set this parameter to a smaller value, a large number of requests are wasted, which results in extra system overheads.

Parameter	Meaning	Description
importSince	The time in seconds based on which to migrate data. Data whose last modified time is greater than the value of this parameter is migrated. The value is of the Integer type.	◦ The timestamp follows the UNIX time format. It is the number of seconds that have elapsed since 00:00:00 January 1, 1970. You can run the date +%s command to obtain the seconds. ◦ The default value is 0, which indicates that all data is to be migrated.
srcType	The source type for synchronization. The value is of the String type. Be aware that the value is case-sensitive.	The following sources are supported: ◦ local: migrates data from a local file to OSS. To specify this option, specify srcPrefix and ignore srcAccessKey, srcSecretKey, srcDomain, and srcBucket. ◦ oss: migrates data from one OSS bucket to another bucket. ◦ qiniu: migrates data from KODO to OSS. ◦ bos: migrates data from BOS to OSS. ◦ ks3: migrates data from KS3 to OSS. ◦ s3: migrates data from Amazon S3 to OSS. ◦ youpai: migrates data from USS to OSS. ◦ http: migrates data from HTTP sources to OSS. ◦ cos: migrates data from COS to OSS. ◦ azure: migrates data from Azure Blob to OSS.
srcAccessKey	The AccessKey ID used to access the source. The value is of the String type.	◦ If srcType is set to oss, qiniu, baidu, ks3, or s3, specify the AccessKey ID used to access the source. ◦ If srcType is set to local or http, ignore this parameter. ◦ If srcType is set to youpai or azure, specify the account username used to access the source.
srcSecretKey	The AccessKey secret used to access the source. The value is of the String type.	◦ If srcType is set to oss, qiniu, baidu, ks3, or s3, specify the AccessKey secret used to access the source. ◦ If srcType is set to local or http, ignore this parameter. ◦ If srcType is set to youpai, specify the password of the operator account used to access the source. ◦ If srcType is set to azure, specify the account key used to access the source.

Parameter	Meaning	Description
srcDomain	The endpoint of the source.	◦ If srcType is set to local or http, ignore this parameter. ◦ If srcType is set to oss, specify the endpoint obtained from the OSS console. The endpoint is a second-level domain without the bucket name. ◦ If srcType is set to qiniu, enter the domain name corresponding to the bucket obtained from the KODO console. ◦ If srcType is set to bos, enter the BOS domain name. Example: `http://bj.bcebos.com` or `http://gz.bcebos.com`. ◦ If srcType is set to ks3, enter the KS3 domain name. Example: `http://kss.ksyun.com`, `http://ks3-cn-beijing.ksyun.com`, or `http://ks3-us-west-1.ksyun.coms`. ◦ If srcType is set to S3, enter the domain name of the region in which your Amazon S3 resources are located. ◦ If srcType is set to youpai, enter the USS domain name such as automatic identification of the optimal path of `http://v0.api.upyun.com`, telecommunication line `http://v1.api.upyun.com`, China Unicom or China Netcom line `http://v2.api.upyun.com`, or China Mobile or China Railcom line `http://v3.api.upyun.com`. ◦ If srcType is set to cos, enter the region in which your COS bucket is located. Example: ap-guangzhou. ◦ If srcType is set to azure, enter the endpoint suffix in the Azure Blob connection string. Example: core.chinacloudapi.cn.
srcBucket	The name of the source bucket or container.	◦ If srcType is set to local or http, ignore this parameter. ◦ If srcType is set to azure, enter the name of the source container. ◦ In other cases, enter the name of the source bucket.

Parameter	Meaning	Description
srcPrefix	The source prefix. The value is of the String type. This parameter is empty by default.	◦ If srcType is set to local, enter the full path that ends with a forward slash (/). If the path contains two or more directory levels, separate each directory with one forward slash (/). Example: c:/example/ or /data/example/. **Notice** Paths such as c:/example//, /data//example/, and /data/example// are invalid. ◦ If srcType is set to oss, qiniu, bos, ks3, youpai, or s3, enter the prefix for objects to be synchronized. The prefix excludes bucket names. Example: data/to/oss/. ◦ To synchronize all files, leave srcPrefix empty.
destAccessKey	The AccessKey ID used to access the destination. The value is of the String type.	To obtain the AccssKey ID, log on to the Alibaba Cloud Management Console.
destSecretKey	The AccessKey secret used to access the destination. The value is of the String type.	To obtain the AccssKey secret, log on to the Alibaba Cloud Management Console.
destDomain	The destination endpoint. The value is of the String type.	To obtain the second-level domain without the bucket name, log on to the Alibaba Cloud Management Console.
destBucket	The destination bucket. The value is of the String type.	The name of the OSS bucket. The name cannot end with a forward slash (/).

Parameter	Meaning	Description
destPrefix	The destination prefix. The value is of the String type. This parameter is empty by default.	◦ The destination prefix. If you retain the default value, the migrated objects are stored in the destination bucket. ◦ To synchronize data to a specified directory in OSS, end the prefix with a forward slash (/). Example: data/in/oss/. ◦ Be aware that OSS object names cannot start with a forward slash (/). Do not start the destination prefix with a forward slash (/). ◦ A local file whose path is in the srcPrefix+relativePath format is migrated to the OSS path in the destDomain/destBucket/destPrefix+relativePath format. ◦ An object in the cloud whose path is in the srcDomain/srcBucket/srcPrefix+relativePath format is migrated to the OSS path in the destDomain/destBucket/destPrefix and relativePath format.
taskObjectCountLimit	The maximum number of files in each Task. The value is of the Integer type. The default value is 10000.	This value specified for this parameter affects the concurrency of jobs to run. In most cases, this parameter is set based on the following formula: Value = Total number of files/Total number of Workers/Number of migration threads. In the preceding formula, the number of migration threads is specified by the workerTaskThreadNum parameter. The maximum value is 50000. If the total number of files is unknown, retain the default value.
taskObjectSizeLimit	The maximum data size in bytes for each task. The value is of the Integer type. The default value is 1 GB.	This value specified for this parameter affects the concurrency of jobs to run. In most cases, this parameter is set based on the following formula: Value = Total data size/Total number of Workers/Number of migration threads. In the preceding formula, the number of migration threads is specified by the workerTaskThreadNum parameter. If the total data size is unknown, retain the default value.
isSkipExistFile	Indicates whether to skip the existing objects during data migration. The value is of the Boolean type.	If this parameter is set to true, ossimport determines whether the objects are skipped based on the size and the last modified time of the objects. If this parameter is set to false, the existing objects are overwritten. The default value is false. The value specified for this parameter is invalid when jobType is set to audit.

Parameter	Meaning	Description
scanThreadCount	The number of threads that scan files in parallel. The value is of the Integer type. ○ Default value: 1. ○ Valid values: 1 to 32	This configuration option is related to file scanning efficiency. If you do not have special requirements, retain the default value.
maxMultiThreadScanDepth	The maximum allowable depth of directories for parallel scanning. The value is of the Integer type. ○ Default value: 1. ○ Valid values: 1 to 16	○ A value of 1 indicates that parallel scanning is performed within top-level directories. ○ If you do not have special requirements, retain the default value. A large value may cause task failures.
appId	The application ID (account number) of COS. The value is of the Integer type.	This parameter is valid when srcType is set to cos.
httpListFilePath	The absolute path of the HTTP list file. The value is of the String type.	○ This parameter is valid when srcType is set to http. When the source is accessed through an HTTP link, you must provide the absolute path of the file that contains the HTTP link. Example: c:/example/http.list. ○ The HTTP link in the file must be divided into two columns separated with spaces, which indicates the prefix and the relative path in OSS after the upload. For example, the c:/example/http.list file can contain two rows: `http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/aa/ bb.jpg` and `http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/cc/dd.jpg`. The object names in OSS after the objects are migrated are in destPrefix+ bb.jpg and destPrefix+cc/dd.jpg formats.

- workers: exclusive to the distributed mode. Each IP address is separated with a line break. Examples:

```
192.168.1.6
192.168.1.7
192.168.1.8
```

- In the preceding configuration, `192.168.1.6` is in the first row, which must be master. `192.168.1.6` is the IP address of the machine where Master, Worker, and TaskTracker are started. Console also runs on this machine.
- Make sure that the username, logon mode, and working directory of each Worker machine in multiple-Worker mode are the same.

Configuration file examples

The following table describes the configuration file of a data migration job in distributed mode. The name of the configuration file in standalone mode is `local_job.cfg`, which contains the same configuration items as those in distributed mode.

Migration type	Configuration file	Description
Migrate local data to OSS	job.cfg	srcPrefix specifies an absolute path that ends with a forward slash (/). Example: `D:/work/oss/data/` or `/home/user/work/oss/data/`.
Migrate data from KODO to OSS	job.cfg	You can leave srcPrefix and destPrefix unspecified. If you want to specify these parameters, end the prefixes with a forward slash (/). Example: `destPrefix=docs/`.
Migrate data from BOS to OSS	job.cfg	You can leave srcPrefix and destPrefix unspecified. If you want to specify these parameters, end the prefixes with a forward slash (/). Example: `destPrefix=docs/`.
Migrate data from Amazon S3 to OSS	job.cfg	For more information, visit AWS service endpoints.
Migrate data from USS to OSS	job.cfg	Set srcAccessKey and srcSecretKey to the username and the password of the operator account.
Migrate data from COS to OSS	job.cfg	Set srcDomain based on V4. Example: `srcDomain=sh`. You can leave srcPrefix unspecified. If you want to specify this parameter, start and end the prefix with a forward slash (`/`). Example: `srcPrefix=/docs/`.
Migrate data from Azure Blob to OSS	job.cfg	Set srcAccessKey and srcSecretKey to the storage account and access key. Set srcDomain to the endpoint suffix in the Azure Blob connection string. Example: `core.chinacloudapi.cn`.

Migration type	Configuration file	Description
Migrate data between buckets in OSS	job.cfg	This method is suitable for data migration between different regions, different storage classes, and objects whose names have different prefixes. We recommend that you deploy your service on ECS and use the domain name for access over the internal network to minimize the traffic cost.

Advanced setting

- Time-specific throttling

  In the *sys.properties* configuration file, workerMaxThroughput(KB/s) specifies the upper throttling limit for Worker. To configure throttling for business such as throttling for the source and network limits, set this parameter to a value smaller than the maximum available bandwidth for the machine based on business requirements. After the modification is complete, restart the service for the modification to take effect.

  In distributed mode, modify the *sys.properties* configuration file in the *$OSS_IMPORT_WORK_DIR/conf* directory for each Worker. Restart the service.

  To implement time-specific throttling, modify the *sys.properties* configuration file as scheduled by using crontab and restart the service for the modification to take effect.

- Modify the number of concurrent tasks
  - In the *sys.properties* configuration file, workerTaskThreadNum specifies the number of concurrent tasks run by Worker. If the network conditions are poor and Worker has to process a large number of tasks, timeout errors are returned. To resolve this issue, modify the configuration by reducing the number of concurrent tasks and restart the service.
  - In the *sys.properties* configuration file, workerMaxThroughput(KB/s) specifies the upper throttling limit for Worker. To configure throttling for business such as throttling for the source and network limits, set this parameter to a value smaller than the maximum available bandwidth for the machine based on business requirements.
  - In the *job.cfg* configuration file, taskObjectCountLimit specifies the maximum number of files in each task. The default value is 10000. This parameter configuration affects the number of tasks. The efficiency of implementing concurrent tasks may degrade if you set this parameter to a small value.
  - In the *job.cfg* configuration file, taskObjectSizeLimit specifies the maximum data size for each task. The default value is 1 GB. This parameter configuration affects the number of tasks. The efficiency of implementing concurrent tasks may degrade if you set this parameter to a small value.

    Notice
    - Before you start your data migration, complete the configurations of the parameters in the configuration files.
    - After you modify parameters in the sys.properties configuration file, restart the local server or the ECS instance on which ossimport is deployed for the modification to take effect.
    - After job.cfg is submitted, parameters in the job.cfg configuration file cannot be modified.

- Data verification without migration

  To specify that ossimport only verifies data without migrating data, set the *job.cfg* or *local_job.cfg* configuration file. Set jobType to *audit* instead of *import*. Configurations of other parameters are the same as those for data migration.

- Incremental data migration mode

  After the migration task starts in incremental data migration mode, data is migrated in intervals. The first data migration task to migrate existing data is started after you submit the job. Then, incremental data is migrated at intervals. The incremental data migration mode is suitable for data backup and synchronization.

  Configure the following configuration items for the incremental data migration mode:

  - In the *job.cfg* configuration file, isIncremental specifies whether to enable the incremental data migration mode. *true* indicates that the incremental data migration mode is enabled. *false* indicates that the incremental data migration mode is disabled. The default value is false.
  - In the *job.cfg* configuration file, incrementalModeInterval indicates the interval at which incremental data migration is implemented. Unit: seconds. The configuration takes effect when you set `isIncremental` to true. The minimum configurable value for incrementalModeInterval is `900`. We recommend that you do not set this parameter to a value smaller than `3600`. If you set this parameter to a smaller value, a large number of requests are wasted, resulting in extra system overheads.

- Filtering conditions for objects to migrate

  You can set filtering conditions to migrate objects that meet specified conditions. ossimport allows you to specify prefixes and last modified time.

  - In the *job.cfg* configuration file, srcPrefix specifies the prefix of the source objects. This parameter is empty by default.
    - If you specify `srcType` as local, enter the local directory path. Enter the full path that ends with a forward slash (/). If the path contains two or more directory levels, separate each directory with a forward slash (/). Example: `c:/example/` or `/data/example/`.
    - If you specify `srcType` as `oss`, `qiniu`, `bos`, `ks3`, `youpai`, or `s3`, enter the name prefix of objects to migrate. Example: `data/to/oss/`. To migrate all objects, leave `srcPrefix` unspecified.
  - In the *job.cfg* configuration file, importSince specifies the last modified time for objects to migrate. Unit: seconds. importSince specifies the timestamp that follows the UNIX time format. It is the number of seconds that have elapsed since 00:00:00 January 1, 1970. You can run the **date +%s** command to obtain the seconds. The default value is 0, which indicates that all data is to be migrated. In incremental data migration mode, this parameter is valid only for the first full migration. In non-incremental mode, this parameter is valid for the entire migration job.
    - If the `LastModified Time` of an object is earlier than `importSince`, the object is not migrated.
    - If the `LastModified Time` of an object is later than `importSince`, the object is migrated.

4.2. Standalone deployment

This topic describes how to deploy ossimport in a standalone method. You can deploy ossimport in the standalone method in Linux or Windows systems.

Context

JDK 1.7 and 1.8 are installed.

Quick start

1. Download ossimport-2.3.5.zip and decompress the file.

   The structure of the decompressed file is as follows:

```
ossimport
├── bin
│ └── ossimport2.jar  # The JAR package that contains the Master, Worker, TaskTracker,
and Console modules.
├── conf
│ ├── local_job.cfg   # The Job configuration file.
│   └── sys.properties  # The configuration file that contains system parameters.
├── console.bat         # The Windows command line utility that is used to run tasks st
ep by step.
├── console.sh          # The command-line tool in Linux used to run tasks step by step
.
├── import.bat          # The script that automatically imports files based on the conf
/local_job.cfg configuration file in Windows. The configuration file contains parameter
s that specify data migration operations such as start, migration, verification, and re
try.
├── import.sh           # The script that automatically imports files based on the conf
/local_job.cfg configuration file in Linux. The configuration file contains parameters
that specify data migration operations such as start, migration, verification, and retr
y.
├── logs                # The directory that contains logs.
└── README.md           # The file that introduces or explains ossimport. We recommend
that you read this file before you use ossimport.
```

2. Edit the following configuration files as needed: *conf/sys.properties* and *conf/local_job.cfg*.

   Do not modify the following items:

   - *workingDir*, *workerUserName*, *workerPassword*, and *privateKeyFile* in *conf/sys.properties*.
   - The name and path of `conf/local_job.cfg` and `jobName` in the configuration file.

   For more information about how to edit configuration files, see Configuration file examples.

   > **Note** Confirm the parameters in *sys.properties* and *local_job.cfg* before submitting a job. The parameters for a job cannot be changed after the job is submitted.

3. Start migration jobs.

   - Double-click the *import.bat* file in Windows.
   - Run **bash import.sh** in Linux.

Running method

In ossimport deployed in the standalone method, data migration jobs are implemented in the following methods:

- One-click import: This method encapsulates all steps into a script. You can implement a data

migration job by following the prompts of the script. If you are a beginner, we recommend that you use this method to implement data migration jobs, as described in the Quick start section.

i. Start one-click import.
   - Double-click the *import.bat* file in Windows.
   - Run **bash import.sh** in Linux.

   > **Note** If the last job that you have run is not complete, ossimport prompts you whether to continue the job from the recorded checkpoint or restart a synchronization job. If you want to start a new data migration job or have modified the source and destination buckets, restart a synchronization job.

ii. In Windows, a new Command Prompt is launched to implement the job and display the logs. In the Command Prompt window that runs ossimport, the status of the job is displayed every 10 seconds. Do not close either of the Command Prompt window during the job. In Linux, the job is implemented in the background.

iii. If a task failed during the job, you are prompted whether to retry. Input *y* to retry, or input *n* to skip and exit.

iv. If a data migration job fails, you can view *master/jobs/local_test/failed_tasks/<tasktaskid>/audit.log* to identify the cause of the failure.

- Step-by-step import: This method performs the following steps to implement a data migration job.

i. Clear jobs with the same name.

   If you have run a job with the same name and want to run the job again, clear the job with the same name first. If you have not run the job or you want to retry the tasks of a failed job, do not run the clear command.
   - In Windows, run **console.bat clean** in the Command Prompt.
   - In Linux, run the **bash console.sh clean** command.

ii. Submit the data migration job.

   You cannot submit jobs with the same name. In this case, clear the job with the same name first. The configuration file of the job to submit is *conf/local_job.cfg*. The default name of the job to submit is `local_test`. Run the following commands to submit the job.
   - In Windows, run **console.bat submit** in the Command Prompt.
   - In Linux, run **bash console.sh submit**.

iii. Start NIC multi-queue.
   - In Windows, run **console.bat start** in the Command Prompt.
   - In Linux, run **bash console.sh start**.

iv. View job status.
   - In Windows, run **console.bat stat** in the Command Prompt.
   - In Linux, run **bash console.sh stat**.

v. Retry failed tasks.

   Tasks may fail because of network issues or other reasons. When you run the retry command, only failed tasks are retried.
   - In Windows, run **console.bat retry** in the Command Prompt.

- In Linux, run **bash console.sh retry**.

vi. Stop the service.

- In Windows, close the *%JAVA_HOME%/bin/java.exe* window.
- In Linux, run **bash console.sh stop**.

Job status and logs

After a job is submitted, the master splits the job into tasks, the workers run the tasks and the tracker collects the task status. After the job is completed, the structure of the ossimport directory is as follows:

```
ossimport
├── bin
│   └── ossimport2.jar    # The JAR package of the ossimport deployed in the standalone met
hod.
├── conf
│   ├── local_job.cfg     # The Job configuration file for the standalone method.
│   └── sys.properties    # The configuration file that contains system parameters.
├── console.sh            # The command-line tool.
├── import.sh             # The script for one-click import.
├── logs
│   ├── import.log        # Migration logs.
│   ├── job_stat.log      # Job status logs.
│   ├── ossimport2.log    # Running logs of ossimport deployed in the standalone method.
│   └── submit.log        # Logs of submitted jobs.
├── master
│   ├── jobqueue                # Jobs that are not split.
│   └── jobs                    # Job status.
│       └── local_test          # Job name.
│           ├── checkpoints     # Checkpoints generated when Master splits jobs into tasks
.
│           │   └── 0
│           │       └── 034DC9DD2860B0CFE884242BC6FF92E7.cpt
│           ├── dispatched      # Tasks dispatched to workers but are not complete.
│           │   └── localhost
│           ├── failed_tasks    # Failed tasks.
│           ├── pending_tasks   # Tasks that are not dispatched.
│           └── succeed_tasks   # Tasks that run successfully.
│               └── A41506C07BF1DF2A3EDB4CE31756B93F_1499744514501@localhost
│                   ├── audit.log    # The running logs of tasks. You can view the logs to
identify error causes.
│                   ├── DONE         # The mark file of successful tasks.
│                   ├── error.list  # Error list of tasks. You can view the errors in the f
ile.
│                   ├── STATUS       # The mark file that indicates task status. The conten
t of this file is Failed or Completed, indicating that the subtask failed or succeeded.
│                   └── TASK         # Description of the tasks.
└── worker      # Status of the task being run by the worker. After the running is complete
d, tasks are managed by the master.
    └── jobs
        └── local_test
            └── tasks
```

> **Notice**
>
> - To view the running information about jobs, view *logs/ossimport2.log* or *logs/import.log*.
> - To know the causes of failed tasks, view *master/jobs/${JobName}/failed_tasks/${TaskName}/audit.log*.
> - To view errors occurred during the task, view *master/jobs/${JobName}/failed_tasks/${TaskName}/error.list*.
> - The preceding log files are for reference only. Do not deploy your services and application based on them.

Common problems about migration failures

- If the files in the source path are modified during upload, an error that contains `SIZE_NOT_MATCH` are recorded in *log/audit.log*. In this case, only the files before modification are uploaded to OSS.
- If the source file is deleted during upload, the migration job may fail.
- If the name of the file to upload does not conform to the naming conventions of OSS (for example, start with a forward slash or be empty), the upload fails.
- If you failed to download the source object from OSS because of network conditions or insufficient permissions, view *logs/ossimport2.log* or *logs/import.log* to identify the cause of the failure.
- The program exits unexpectedly and the job status is Abort. If this happens, contact after-sales technical support.

4.3. Distributed deployment

This topic describes how to deploy ossimport in a distributed method. You can deploy ossimport in the distributed method only in Linux.

Download

Download the tool for distributed deployment: ossimport-2.3.5.tar.gz. Download the tool to a local directory and run the command `tar -zxvf ossimport-2.3.5.tar.gz -C $HOME/ossimport` to decompress the file. The file structure of the decompressed tool is as follows:

```
ossimport
├── bin
│   ├── console.jar     # The JAR package for the Console module.
│   ├── master.jar      # The JAR package for the Master module.
│   ├── tracker.jar     # The JAR package for the Tracker module.
│   └── worker.jar      # The JAR package for the Worker module.
├── conf
│   ├── job.cfg         # The Job configuration file template.
│   ├── sys.properties  # The configuration file that contains system parameters.
│   └── workers         # The list of Workers.
├── console.sh          # The command-line tool. Currently, only Linux is supported.
├── logs                # The directory that contains logs.
└── README.md           # The file that introduces or explains ossimport. We recommend that
you read this file before you use ossimport.
```

Note:

- OSS_IMPORT_HOME: The root directory of ossImport. By default the directory is the `$HOME/ossimport` in the decompress command. You can also run the `export OSS_IMPORT_HOME=<dir>` command or modify the system configuration file `$HOME/.bashrc` to set the directory. We recommend that you keep the default value.
- OSS_IMPORT_WORK_DIR: The working directory of ossimport. You can specify the directory by setting the value of `workingDir` in `conf/sys.properties`. We recommend that you set the value to `$HOME/ossimport/workdir`.
- Specify absolute paths for OSS_IMPORT_HOME or OSS_IMPORT_WORK_DIR, such as `/home/<user>/ossimport` or `/home/<user>/ossimport/workdir`.

Parameter

The distributed deployment of ossimport is performed based on three configuration files: `conf/sys.properties`, `conf/job.cfg`, and `conf/workers`.

- `conf/job.cfg`: The configuration file template used to configure jobs in distributed mode. Modify the values according to the actual parameters before data migration.
- `conf/sys.properties`: The configuration file used to configure system operating parameters, such as working directory and worker.
- `conf/workers`: : The worker list.

> **Notice**
> - Confirm the parameters in `sys.properties` and `job.cfg` before submitting a job. The parameters for a job cannot be changed after the job is submitted.
> - Configure `workers` before you start the service. The configuration in this file cannot be added or deleted after the service is started.

Running

- Perform migration tasks

  The distributed deployment of ossimport is performed in the following steps:

  - Deploy the service. Run the bash console.sh deploy command in Linux.

    > **Note** Ensure the configuration files such as conf/job.cfg and conf/workers are properly configured before you deploy the service.

  - Clear jobs with the same name. If you have run a job with the same name and want to run the job again, clear the job with the same name first. If you have not run the job or you want to retry the tasks of a failed job, do not run the clear command. To clear jobs with the same name, run the `bash console.sh clean job_name` command in Linux.

- Submit the data migration job. You cannot submit jobs with the same name. In this case, run the `clean` command to clear the jobs. To submit a job, you must specify the configuration file of the job. You can create the configuration file of a job by modifying the `conf/job.cfg` template. Run `bash console.sh submit [job_cfg_file]` in Linux to submit a job that uses job_cfg_file as its configuration file. In the command, `job_cfg_file` is optional and is set to `$OSS_IMPORT_HOME/conf/job.cfg` by default. By default, the value of `$OSS_IMPORT_HOME` is the path where `console.sh` is located.
- Start the service. Run the `bash console.sh start` command in Linux.
- View job status. Run the `bash console.sh stat` command in Linux.
- Retry failed tasks. Tasks may fail because of network issues or other reasons. When you run the retry command, only failed tasks are retried. Run the `bash console.sh retry [job_name]` command in Linux. In the command, `job_name` is an optional parameter that specifies the jobs in which failed tasks need to retry. If this parameter is not specified, failed tasks of all jobs are retried.
- Stop the service. Run the `bash console.sh stop` command in Linux.

Note:

- If an error occurs because of incorrect parameters in a command, `bash console.sh` prompts you the correct command format.
- We recommend that you specify absolute paths for the paths in configuration files and submitted jobs.
- The configurations in `job.cfg`

> **Note** cannot be modified after the file is submitted. Confirm the items in the file before you submit the file.

- Common causes of job failures
  - If the files in the source path are modified during upload, an error that contains `SIZE_NOT_MATCH` are recorded in `log/audit.log`. In this case, only the files before modification are uploaded to OSS.
  - If the source file is deleted during upload, you may fail to download the file.
  - If the name of the file to upload does not conform to the naming conventions of OSS (for example, start with a forward slash or be empty), the upload fails.
  - Fail to download the source file from OSS.
  - The program exits unexpectedly and the job state is Abort. If this happens, contact after-sales technical support.
- Job status and logs

  After a job is submitted, the master splits the job into tasks, the workers run the tasks and the tracker collects the task status. After a job is completed, the structure of the workdir directory is as follows:

```
workdir
├── bin
│ ├── console.jar     # The JAR package for the Console module.
│ ├── master.jar      # The JAR package for the Master module.
│ ├── tracker.jar     # The JAR package for the Tracker module.
│ └── worker.jar      # The JAR package for the Worker module.
```

```
├── conf
│ ├── job.cfg         # The Job configuration file template.
│ ├── sys.properties  # The configuration file that contains system parameters.
│ └── workers         # The list of Workers.
├── logs
│   ├── import.log      # Migration logs.
│   ├── master.log      # Master logs.
│   ├── tracker.log     # Tracker logs
│ └── workers         # Worker logs.
├── master
│   ├── jobqueue                # Jobs that are not split.
│   └── jobs                    # Job status.
│       └── xxtooss             # Job names.
│           ├── checkpoints     # Checkpoints generated when Master splits jobs into tas
ks.
│           │   └── 0
│           │       └── ED09636A6EA24A292460866AFDD7A89A.cpt
│           ├── dispatched      # Tasks dispatched to workers but are not complete.
│           │   └── 192.168.1.6
│           ├── failed_tasks    # Failed tasks.
│           │   └── A41506C07BF1DF2A3EDB4CE31756B93F_1499348973217@192.168.1.6
│           │       ├── audit.log     # The running logs of tasks. You can view the logs
to identify error causes.
│           │       ├── DONE          # The mark file of successful tasks. If the task fa
ils, the content is empty.
│           │       ├── error.list    # Error list of tasks. You can view the errors in t
he file.
│           │       ├── STATUS        # The mark file that indicates task status. The con
tent of this file is Failed or Completed, indicating that the task failed or succeeded.
│           │       └── TASK          # Description of the tasks.
│           ├── pending_tasks   # Tasks that are not dispatched.
│           └── succeed_tasks   # Tasks that run successfully.
│               └── A41506C07BF1DF2A3EDB4CE31756B93F_1499668462358@192.168.1.6
│                   ├── audit.log    # The running logs of tasks. You can view the logs t
o identify error causes.
│                   ├── DONE         # The mark file of successful tasks.
│                   ├── error.list   # Error list of tasks. If the tasks are successful ,
the error list is empty.
│                   ├── STATUS       # The mark file that indicates task status. The cont
ent of this file is Failed or Completed, indicating that the subtask failed or succeeded.
│                   └── TASK         # Description of the tasks.
└── worker  # Status of the task being run by the worker. After running, tasks are manage
d by the master.
    └── jobs
        ├── local_test2
        │   └── tasks
        └── local_test_4
            └── tasks
```

Notice

- To view the running information about jobs, view `logs/import.log`.
- To know the causes of failed tasks, view `master/jobs/${JobName}/failed_tasks/${TaskName}/audit.log`.
- To view errors occurred during the task, view `master/jobs/${JobName}/failed_tasks/${TaskName}/error.list`.
- The preceding log files are for reference only. Do not deploy your services and application based on them.

Common errors and troubleshooting

For more information about common errors and troubleshooting, see FAQ.

4.4. Troubleshooting

This topic describes the causes and solutions for common problems you may encounter when you use ossimport.

Note All ossimport commands mentioned in this topic are shortened and the complete form must be used in practice.

- Add **console.bat** to Windows commands. For example, change **submit** to **console.bat submit**.
- Add **bash console.sh** to Linux commands. For example, change **submit** to **bash console.sh submit**.

Common problems about migration failures

If a migration job fails, we recommend that you view the migration failure log to identify the cause. After resolving the problems, you can run the **retry** command to migrate the files again. The path of the migration failure log is *master/jobs/${JobName}/failed_tasks/${TaskName}/audit.log*.

- The job state is displayed as failed when you run the **stat** command.

```
[root@l-other-server3.bit.prod.aws.dm import]# bash console.sh stat
================ Process Status =================
=========================================
[2019-04-26 08:49:47 0026][JobTracker][ INFO ] - [2019-04-26 08:49:47] [JobTracker] [INFO ] -
[2019-04-26 08:49:47] [JobTracker] [INFO ] - jobName:[illegible]-bi-s3-20190419
[2019-04-26 08:49:47] [JobTracker] [INFO ] - JobState:RUNNING
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Pending Task Count:0
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Dispatched Task Count:1624
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Succeed Task Count:787
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Failed Task Count:80
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Is Scan Finished:true
[2019-04-26 08:49:47] [JobTracker] [INFO ] - =========================================
```

  Solution: Run the **stat** command to view the job state. If JobState is failed, the migration job fails. After the migration job is complete, run the **retry** command to migrate the files again.

- Some files fail to be migrated and retrying the migration also fails.

  Solution:

i. View the list of failed files from *master/jobs/${JobName}/failed_tasks/${TaskName}/error.list* to obtain their relative paths.

ii. Check whether you are authorized to access these files, whether these files are deleted, whether they are symbolic links, and whether the file names are garbled.

iii. After the preceding problems are resolved, run the **retry** command to migrate the files again.

- "The bucket you are attempting to access must be addressed using the specified endpoint" is displayed in the migration failure log.

```
Exception:com.aliyun.oss.OSSException: The bucket you are attempting to access must be ad
dressed using the specified endpoint. Please send all future requests to this endpoint.
<Error>
  <Code>AccessDenied</Code>
  <Message>The bucket you are attempting to access must be addressed using the specified
endpoint. Please send all future requests to this endpoint. </Message>
  <RequestId>56EA98DE815804**21B23EE6</RequestId>
  <HostId>my-oss-bucket.oss-cn-qingdao.aliyuncs.com</HostId>
  <Bucket>my-oss-bucket</Bucket>
  <Endpoint>oss-cn-hangzhou.aliyuncs.com</Endpoint>
</Error>
```

Analysis: The value of srcDomain or destDomain is invalid. Enter valid endpoints based on the endpoints listed in Regions and endpoints.

- "The request signature we calculated does not match the signature you provided" is displayed in the migration failure log.

```
Exception:com.aliyun.oss.OSSException: The request signature we calculated does not match
the signature you provided. Check your key and signing method.
[ErrorCode]: SignatureDoesNotMatch
[RequestId]: xxxxxxx
[HostId]: xxx.oss-cn-shanghai.aliyuncs.com
```

Analysis: The values of destAccessKey and destSecretKey are invalid. Enter a valid AccessKey pair.

- "The bucket name "xxx/xx" is invalid" is displayed in the migration failure log.

```
java.lang.IllegalArgumentException: The bucket name "xxx/xx" is invalid. A bucket name mu
st: 1) be comprised of lower-case characters, numbers or dash(-); 2) start with lower cas
e or numbers; 3) be between 3-63 characters long.
```

Analysis: Check whether the value of destBucket is valid. The bucket name cannot contain forward slashes (/) and paths.

- "Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 timed out" is displayed in the migration failure log.

```
Unable to execute HTTP request: Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 ti
med out
[ErrorCode]: ConnectionTimeout
[RequestId]: Unknown
```

Analysis: The connection timeout error is returned because the configuration file uses the internal endpoint of OSS, but the device that was used to migrate data is not an ECS instance or is not an ECS instance that is in the same region as OSS instances. The internal endpoint supports access only from ECS instances that are located in the same region as OSS instances.

Solution:

- Set the domain name to a public endpoint in the configuration file. Delete the migration job and submit the job again.
- Use an ECS instance that is located in the same region as OSS instances to implement the migration job.

- "The specified bucket is not valid" is displayed in the migration failure log.

```
com.aliyun.oss.OSSException: The specified bucket is not valid.
[ErrorCode]: InvalidBucketName
[RequestId]: 57906B4DD0EBAB0FF553D661
[HostId]: you-bucket.you-bucketoss-cn-hangzhou-internal.aliyuncs.com
```

Analysis: The value of destDomain must be the endpoint of the region where the bucket is located, rather than the second-level domain name that contains the bucket name. For example, if the bucket is located in China (Beijing), enter oss-cn-beijing.aliyuncs.com. For more information, see Configuration file examples.

- "Unable to execute HTTP request: The Difference between ... is too large" is displayed in the migration failure log.

```
Unable to execute HTTP request: The Difference between the request time and the current t
ime is too large.
[ErrorCode]: RequestTimeTooSkewed
[RequestId]: xxxxxxx
```

Analysis: This error may result from either of the following reasons.

- The difference between the system time of the machine or device sending the request and the system time adopted by OSS exceeds 15 minutes. This is the most common situation.
- A large number of requests are sent at the same time, resulting in a high CPU utilization and slow concurrent uploads.

Solution:

- Set the system time of the machine or device that is sending the request to the same time adopted by OSS.
- If the error is caused by high concurrency, you can reduce the number of requests that are sent at a time. You can set workerTaskThreadNum of the *sys.properties* file to a smaller value.

- "No route to host" is displayed in the migration failure log.

  Analysis: This error is returned when the network connection fails because of local firewalls or iptables.

  Solution: Run the ping command to check whether the network connection between the source and destination instances is normal.

  - If the network connection is normal, you can check if there are restrictions configured by the computer firewalls and local firewalls. You can also try disabling the firewall.
  - If the network connection is abnormal, troubleshoot the problem and try again.

- "Unknown http list file format" is displayed in the migration failure log when you migrate files over HTTP.

  Analysis: This error is returned when the format of the specified HTTP list file or its content is invalid.

Solution:

- You can convert the format of files that are copied from a different operating system by using commands such as mac2unix and doc2unix in Linux and tools such as notepad in Windows.
- If the format of the HTTP list file content is invalid, change the content to the valid format. For more information about the format of the HTTP list file content, see the "Configuration files" section in Architectures and configurations.

- "The object key "/xxxxx.jpg" is invalid" is displayed in the migration failure log.

```
Exception:java.lang.IllegalArgumentException: The object key "/xxxxx.jpg" is invalid. An
object name should be between 1 - 1023 bytes long when encoded as UTF-8 and cannot contai
n LF or CR os unsupported chars in XML1.0, and cannot begin with "/" or "\".
```

Solution:

- Check whether srcPrefix is set to a folder but its value does not end with a forward slash (/). If you set the value to a folder, it must end with a forward slash (/).
- Check whether the value of destPrefix starts with a forward slash (/) or a backslash (\). The value of destPrefix cannot start with a forward slash (/) or a backslash (\).

Common problems when running a migration job

If a problem occurs during the migration, you can first view the running log. When ossimport is deployed in standalone mode, the running log file path is *logs/ossimport2.log*. When ossimport is deployed in distributed mode, the running log file path is *logs/import.log*.

- "UnsupportedClassVersionError" is returned when a command is run.

```
Exception in thread "main" java.lang.UnsupportedClassVersionError: com/aliyun/ossimport2/
OSSImport2 : Unsupported major.minor version 51.0
        at java.lang.ClassLoader.defineClass1(Native Method)
        at java.lang.ClassLoader.defineClassCond(ClassLoader.java:631)
        at java.lang.ClassLoader.defineClass(ClassLoader.java:615)
        at com.simontuffs.onejar.JarClassLoader.defineClass(JarClassLoader.java:693)
        at com.simontuffs.onejar.JarClassLoader.findClass(JarClassLoader.java:599)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
        at com.simontuffs.onejar.Boot.run(Boot.java:300)
        at com.simontuffs.onejar.Boot.main(Boot.java:159)
```

Analysis: The Java version is earlier than the required version. Update it to version 1.7 or 1.8.

- "InvocationTargetException" is returned when you use the **submit** command to submit a job.

```
Exception in thread "main" java.lang.reflect.InvocationTargetException
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
        at java.lang.reflect.Method.invoke(Method.java:497)
        at com.simontuffs.onejar.Boot.run(Boot.java:306)
        at com.simontuffs.onejar.Boot.main(Boot.java:159)
Caused by: java.lang.NullPointerException
        at com.aliyun.ossimport2.config.JobConfig.load(JobConfig.java:44)
        at com.aliyun.ossimport2.OSSImport2.doSubmitJob(OSSImport2.java:289)
        at com.aliyun.ossimport2.OSSImport2.main(OSSImport2.java:120)
        ... 6 more
```

Analysis: This error is returned because the system identifies that some configuration items are deleted or commented out from the configuration file.

Solution: Restore the configuration items that are deleted or commented out from the configuration file. Leave configurations items unspecified after the equal sign (=) instead of deleting them. For configuration file examples, see Configuration file examples.

- "com.aliyun.oss.ClientException: Unknown" is displayed in the running log.

```
com.aliyun.oss.ClientException: Unknown
[ErrorCode]: NonRepeatableRequest
[RequestId]: Cannot retry request with a non-repeatable request entity.  The cause lists
the reason the original request failed.
```

Analysis: "com.aliyun.oss.ClientException: Unknown" or "SocketTimeoutException" error is returned when the server is at full bandwidth. In this case, ossimport automatically retries. If the job still fails, run the **retry** command to migrate the files again.

- "Too many open files" is displayed in the running log in Linux.

  Solution: Run the **ulimit -n** command to view the limit of handles in Linux.

  - If the handle limit is greater than 10,000, use the **ulimit -n 65536** command to increase the value and restart the process.
  - If handle values are greater than 100,000, use the **sudo losf -n** command to check which processes have enabled the handles. You can evaluate these processes and kill the unnecessary processes to release the handles.

- The migration job exits in seconds after starting in Windows

  Analysis:

  - Java is not installed or a Java version earlier than 1.7 is used.
  - Configuration file error.

  Solution:

  - Install the supported Java version.
  - Follow examples to edit configuration files. For more information about configuration file examples, see Configuration file examples.

- After a job is submitted by using the **submit** command, "no jobs is running or finished" is displayed when you use the **stat** command to view the job status.

```
bash console.sh stat
[WARN]   List files dir not exist : /home/<user>/ossimport/workdir/master/jobs/
no jobs is running or finished.
```

Analysis: A job starts running when the service is started and the job is submitted. In this case, you can use the **stat** command to view the job status.

- If you only submitted the job by using the **submit** command, but did not start the service by using the **start** command, "no jobs is running or finished" is displayed. You must use the **start** command to start the service.
- If the service is started and the job has just been submitted, Master must scan the file list first. Because the job has not been generated and distributed at this time, this error is displayed in the logs.
- If the error still occurs after the service has been started and the task has been submitted for a period of time, check whether the process exits unexpectedly after it starts. When ossimport is deployed in standalone mode, view the log file from *logs/ossimport2.log*. When ossimport is deployed in distributed mode, view the log file from *logs/ossimport.log*. Find the cause of the exception, resolve it, and then start the service process.

- "scanFinished: false " is displayed when you use the **stat** command to check the job status.

  Solution: Check whether the total number of tasks is increasing.

  - If the total number of tasks is increasing, the file list of the migration job is listing new files. It is normal that this error is returned in this case.
  - If the incremental data migration mode is enabled, scanFinished is not true, and the number of tasks does not change, ossimport scans the file list at specified intervals to check for new or modified files.
  - If the incremental data migration mode is disabled and the number of tasks does not increase, check the running log for exceptions. When ossimport is deployed in standalone mode, view the log file from *logs/ossimport2.log*. When ossimport is deployed in distributed mode, view the log file from *logs/ossimport.log*. Find the cause of the exception, resolve it, and then start the service process.

- The process in Linux does not respond, but no exception is output in logs

  Analysis: If the system has less than 2 GB of available memory, the process may be killed due to insufficient memory. You can check whether the dmesg log contains records about the processes that were killed because of insufficient memory.

- How do I restart a service after the process stops responding or is killed?

  All submitted jobs have breakpoint records. If you do not clear the original job by using the **clean** command, you can directly use the **start** command to start the service without submitting the job again.

- How do I upload files whose names are garbled to OSS in Linux?

  Solution:

  i. Check the encoding format of the garbled names.

  ii. Use the **export LANG="<your file name encode>"** command to parse these encoded names.

  iii. Use the **clean** command to clear the original job, and then use the **submit** command to submit the job again.

- "java.nio.file.AccessDeniedException" is returned when the service is started.

  Analysis: You are not authorized to access the configuration file.

  Solution: Set the access permission of the configuration file to public read, or log on to Linux as an administrator to start the service.

- Task Counts are displayed as 0, but JobState is displayed as succeed.

  Description: Pending Task Count and Diamond Task Count are both 0, but JobState is SUCCEED.

```
[2015-12-28 16:12:35]   [INFO]  JobName:dir_data
[2015-12-28 16:12:35]   [INFO]  Pending Task Count:0
[2015-12-28 16:12:35]   [INFO]  Dispatched Task Count:0
[2015-12-28 16:12:35]   [INFO]  Succeed Task Count:0
[2015-12-28 16:12:35]   [INFO]  Failed Task Count:0
[2015-12-28 16:12:35]   [INFO]  Is Scan Finished:true
[2015-12-28 16:12:35]   [INFO]  JobState:SUCCEED
```

  Analysis:

  - Files cannot be listed because the value of srcPrefix is invalid.
  - Folders specified by srcPrefix do not have files. Folders cannot be uploaded to OSS because the concept of folders is simulated by OSS.

  Solution: Set a valid value for the srcPrefix parameter and make sure there are available files in the folder specified by srcPrefix.

- "InvocationTargetException" is returned when you submit the job.

```
submit job:/disk2/ossimport2/local_job.cfg
Exception in thread "main" java.lang.reflect.InvocationTargetException
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j
ava:43)
        at java.lang.reflect.Method.invoke(Method.java:606)
        at com.simontuffs.onejar.Boot.run(Boot.java:306)
        at com.simontuffs.onejar.Boot.main(Boot.java:159)
Caused by: java.lang.NullPointerException
        at com.aliyun.ossimport2.OSSImport2.doSubmitJob(OSSImport2.java:289)
        at com.aliyun.ossimport2.OSSImport2.main(OSSImport2.java:120)
        ... 6 more
```

  Analysis: This error is returned because the configuration file or configuration file path is invalid.

  Solution:

  i. Configure the workingDir configuration item in the *conf/sys.properties* file properly.

  ii. If the configuration file is valid, check the configuration file path.

- Source files for synchronization do not exist.

  Analysis: Master first lists the files and then migrates data based on the list. If files are deleted from the source after listing, they cannot be migrated. The deleted files are skipped during migration, and the names are listed in the error list.

- The running status of the migration job is inconsistent with the configuration file.

Description: The job configuration file is valid, but the job running status is inconsistent with the job configuration file during the migration.

Analysis: If a migration job has been submitted, modifications to the job configuration file do not take effect after the job is suspended.

Solution: Clear the previous job by using the **clean** command. After the job configuration file is modified, submit the job again.

Common problems about migrating data from UPYUN Storage Service (USS) to OSS

- The number of jobs is always 0.

  Analysis: View the job running log first.

  ```
  [2016-07-21 10:21:46] [INFO] [name=YoupaiList, totalRequest=1729925, avgLatency=38,
            recentLatency=300000]
  ```

  - If the value of `recentLatency` in the running log is smaller than or equal to 30000, the files are listed properly. It can take more than 30 seconds (timeout) for files to be listed on USS in most cases. Files listed within 30 seconds are returned. This problem disappears when all files are listed.
  - If the value of `recentLatency` is small, this problem occurs because the account password is invalid. USS only returns null instead of error results when SDK errors occur. You can obtain the error code returned by USS for troubleshooting only by capturing packets.

- How to specify `srcAccessKey` and `srcSecretKey`.

  Enter the operator account and password of USS.

- HTTP status code 429 keeps appearing.

  USS has specified the SDK access interval. If too many requests are sent over a short period of time, the system limits the number of requests to be handled. We recommend that you contact the customer service of USS to remove the restrictions. ossimport retries in this case.

Common problems

- After the migration job is complete, the data volume displayed in the OSS console is smaller than the source data volume.

  Description: The bucket size does not change in the OSS console after all the migration job is complete, or the data sizes calculated by using the **du** command vary greatly in Linux.

  Analysis:

  - Bucket sizes in OSS console are updated with a delay of one to two hours.
  - The **du** command in Linux counts the block size, which is larger than the file size. You can use the **ls -lR <absolute folder path> | grep "\-rw" | awk '{sum+=$5}END{print sum}'** command to count the actual size of the local folder.

- The error message such as unknown command "java" or unknown command "nohup" is returned when you run commands in Linux.

  Analysis: The command that you want to run is not installed. Use the **yum**, **apt-get**, or `zypper` command to install the corresponding command based on operating systems.

- Can I use `srcPrefix` in the configuration file to specify files?

You can only use `srcPrefix` to specify folders or prefixes.

- Can I configure proxies for ossimport?

  No. ossimport does not support proxies.

- Why am I charged for migrating data within OSS?

  If you migrate data by using an internal endpoint, you are charged based on the number of requests. No traffic fee is charged.

- Will files that are deleted from the local folder be deleted from OSS if the incremental data migration mode is enabled?

  No. If the incremental data migration mode is enabled, files that are deleted from the local folder will not be deleted from OSS.

- Why are there new files that are not synchronized when the incremental data migration mode is enabled?

  In incremental data migration mode, the last modification time of a file is used to determine whether the file is new. Linux mv command, Windows commands such as cp, mv, and rsync with `-t` or `-a` parameters do not modify the last modification time of files. Files modified by using these commands are not scanned for and therefore are ignored during migration.

- Can file permissions be migrated when files are migrated to OSS?

  No. After the migration is complete, you can use the **set-meta** command of ossutil to modify the permissions. For more information, see set-meta.

5.ossfs

5.1. Overview

ossfs allows you to attach Object Storage Service (OSS) buckets as local file systems in Linux. Then, you can manage OSS objects in the same manner as you manage local files.

Runtime environment

ossfs is developed based on the file system in userspace (fuse), and can only be run on machines that support fuse. OSS provides installation packages for Ubuntu and CentOS. To run the packages in other environments, use source code to build the required program.

You can use ossfs over the Alibaba Cloud internal network or over the Internet. To improve the speed and stability of access, we recommend that you use an internal endpoint within the internal network environment.

Install ossfs

- For more information about how to download and install ossfs, see Installation.
- For more information how to configure ossfs, see Advanced configurations.

Features

ossfs is built based on S3FS and incorporates all S3FS features, including:

- Supports most features of the POSIX file system. For example, you can upload and download objects and directories, and set user permissions.
- Uses multipart upload and resumable upload of OSS to upload objects by default.
- Supports MD5 verification to ensure data integrity.

Usage notes

- You cannot attach a bucket of the Archive storage class.
- ossfs is unsuitable for scenarios that require highly concurrent read and write operations.
- If you use ossfs to edit an object in OSS, the object is uploaded again.
- When you perform metadata-related operations such as **list directory**, the performance of ossfs is compromised because you must remotely access the OSS server.
- Errors may occur if you rename an object or a directory. Operation failures may cause data inconsistency between OSS and local disks.
- If an OSS bucket is attached to multiple clients, you are responsible for maintaining data consistency. We recommend that you schedule the time for your users to use objects to prevent multiple clients from writing to the same object at the same time.
- Hard links are not supported.
- If an upload is interrupted before the object is completely uploaded, the uploaded portion is stored as parts in an OSS bucket. If you no longer need these parts, you can use the following methods to delete them to avoid additional storage fees.
  - Manually delete parts. For more information, see Manage parts.
  - Use lifecycle rules to automatically delete parts. For more information, see Lifecycle rules based on the last modified time.

Notice The performance and features of ossfs are not as good as those of local file systems because data must be synchronized to the cloud over networks. Exercise caution when you want to use ossfs for applications that have high I/O requirements, such as databases.

5.2. Installation

ossfs allows you to mount Object Storage Service (OSS) buckets as local file systems on Linux. Then, you can manage OSS objects in the same manner as you manage local files.

Runtime environment

We recommend that you run ossfs in the following environments:

- Linux-based operating systems
  - CentOS 7.0 or later
  - Ubuntu 14.04 or later
- Fuse 2.8.4 or later

Download URLs

Linux distribution	Download
Ubuntu 18.04 (x64)	ossfs_1.80.6_ubuntu18.04_amd64.deb
Ubuntu 16.04 (x64)	ossfs_1.80.6_ubuntu16.04_amd64.deb
Ubuntu 14.04 (x64)	ossfs_1.80.6_ubuntu14.04_amd64.deb
CentOS 8.0 (x64)	ossfs_1.80.6_centos8.0_x86_64.rpm
CentOS 7.0 (x64)	ossfs_1.80.6_centos7.0_x86_64.rpm

The preceding table provides the download URLs of ossfs installation packages for common operating systems. If your operating system is not included in the table, install ossfs by compiling the source code. For more information about the source code, visit GitHub ossfs.

Notice When you copy the URLs and use them in the wget command to download ossfs, delete `?spm=xxxx` from the URLs.

Installation

1. Download the installation package.

   For example, run the following command to download the ossfs installation package for CentOS 7.0 (x64):

   ```
   wget http://gosspublic.alicdn.com/ossfs/ossfs_1.80.6_centos7.0_x86_64.rpm
   ```

2. Install ossfs.
   - Ubuntu

For example, you can run the following command to install ossfs in Ubuntu 16.04 (x64):

```
sudo apt-get update
sudo apt-get install gdebi-core
sudo gdebi ossfs_1.80.6_ubuntu16.04_amd64.deb
```

- CentOS

  For example, you can run the following command to install ossfs in CentOS 7.0 (x64):

  ```
  sudo yum install ossfs_1.80.6_centos7.0_x86_64.rpm
  ```

  If your client uses Yellowdog Updater, Modified (YUM) to install the Red-hat Package Manager (RPM) package, dependencies may fail to be downloaded when the client node network environment is not suitable. To resolve this issue, you can use YUM to download the dependencies over the normal network to a node that runs the same operating system version, and then copy the dependencies to the required node. For example, ossfs runs based on fuse 2.8.4 or later. You can run the following command to download the latest version of fuse from the YUM source to your local device:

  ```
  sudo yum install --downloadonly --downloaddir=./ fuse
  ```

  Notice To download other dependencies, replace fuse with the name of the required package.

3. Configure access information of the account.

   Store the bucket name and the AccessKey pair that can be used to access the bucket in the */etc/passwd-ossfs* object. We recommend that you set the object permission to 640.

   ```
   echo BucketName:yourAccessKeyId:yourAccessKeySecret > /etc/passwd-ossfs
   chmod 640 /etc/passwd-ossfs
   ```

4. Mount the bucket to the specified directory.

   ```
   ossfs BucketName mountfolder -o url=Endpoint
   ```

   For example, you can run the following command to mount a bucket named `bucket-test` to the `/tmp/ossfs` directory.

   ```
   echo bucket-test:LTAIbZcdVCmQ****:MOk8x0y9hxQ31coh7A5e2MZEUz**** > /etc/passwd-ossfs
   chmod 640 /etc/passwd-ossfs
   mkdir /tmp/ossfs
   ossfs bucket-test /tmp/ossfs -o url=http://oss-cn-hangzhou.aliyuncs.com
   ```

   Notice If you use an Alibaba Cloud Elastic Compute Service (ECS) instance to provide ossfs services, you can use the internal endpoint of the bucket. In the preceding example, you can set the endpoint of OSS to `oss-cn-hangzhou-internal.aliyuncs.com` to minimize traffic fees. For more information about the internal endpoints of OSS, see Regions and endpoints.

5. If you no longer want to mount the bucket, you can run the following command to unmount it:

   ```
   fusermount -u /tmp/ossfs
   ```

For more information, visit GitHub ossfs.

Version history

For more information about the version history of ossfs, visit GitHub ChangeLog.

5.3. Advanced configurations

This topic describes how to configure ossfs.

Prerequisites

ossfs is installed. For more information about how to install ossfs, see Installation.

Configure the account information

When you use ossfs to access Object Storage Service (OSS) buckets, you must configure your account information including your AccessKey ID and AccessKey secret. The account information must be written to the account configuration file in a specific format. ossfs obtains the account information in the `$bucket_name:$access_key_id:$access_key_secret` format from the account configuration file.

The default path for the account configuration file is */etc/passwd-ossfs*. You can also use the -opasswd_file=passwd-path option to specify a configuration file. The permission of a configuration file in the default path may be 640, while that of a configuration file in other paths must be 600.

- An account configuration file can contain records for multiple accounts. Each line indicates the information of one account. When ossfs is used to attach a bucket, ossfs matches the bucket name with the correct account.

  Configuration examples:

  ```
  echo bucket-test-1:AAAIbZcdVCmQ****:AAA8x0y9hxQ31coh7A5e2MZEUz**** > /etc/passwd-ossfs
  echo bucket-test-2:BBBIbZcdVCmQ****:BBB8x0y9hxQ31coh7A5e2MZEUz**** >> /etc/passwd-ossfs
  chmod 640 /etc/passwd-ossfs
  mkdir /tmp/ossfs-1
  mkdir /tmp/ossfs-2
  ossfs bucket-test-1 /tmp/ossfs-1 -ourl=http://oss-cn-hangzhou.aliyuncs.com
  ossfs bucket-test-2 /tmp/ossfs-2 -ourl=http://oss-cn-hangzhou.aliyuncs.com
  ```

- If you want to attach multiple buckets, you can choose to write the information of every account to one account configuration file, or write the information of different accounts to different account configuration files. You can use the -opasswd_file=xxx option to select the account configuration file.

  Configuration examples:

  ```
  echo bucket-test-3:CCCIbZcdVCmQ****:CCC8x0y9hxQ31coh7A5e2MZEUz**** > /etc/passwd-ossfs-3
  chmod 600 /etc/passwd-ossfs-3
  mkdir /tmp/ossfs-3
  ossfs bucket-test-3 /tmp/ossfs-3 -ourl=http://oss-cn-hangzhou.aliyuncs.com -opasswd_file=
  /etc/passwd-ossfs-3
  echo bucket-test-4:DDDIbZcdVCmQ****:DDD8x0y9hxQ31coh7A5e2MZEUz**** > /etc/passwd-ossfs-4
  chmod 600 /etc/passwd-ossfs-4
  mkdir /tmp/ossfs-4
  ossfs bucket-test-4 /tmp/ossfs-4 -ourl=http://oss-cn-hangzhou.aliyuncs.com -opasswd_file=
  /etc/passwd-ossfs-4
  ```

Use instance RAM roles

In Elastic Compute Service (ECS), you can use ossfs based on instance Resource Access Management (RAM) roles. You can bind a RAM role to an ECS instance to access OSS from the instance by using Security Token Service (STS) temporary credentials. STS temporary credentials are automatically generated and updated. Applications can obtain the STS temporary credentials by using the instance metadata URL. RAM helps protect the security of your AccessKey pair and facilitates fine-grained permission control and management. For more information about instance RAM roles, see Overview.

The following section describes how to use ossfs by using the EcsRamRoleOssTest instance RAM role:

1. Create a RAM role named EcsRamRoleOssTest.

   For more information about how to create a RAM role, see Step 1: Create an instance RAM role.

2. Grant the RAM role permissions to access OSS resources.

   For more information about how to grant permissions to a RAM role, see Grant permissions to a RAM role. In this example, the RAM role is granted the AliyunOSSReadOnlyAccess permission. You can customize the permission. For more information, see Create a custom policy.

3. Bind an ECS instance to the RAM role.

   For more information, see Step 3: Attach the RAM role to an ECS instance.

4. Use ossfs based on the instance metadata URL.

   i. Log on to the ECS instance.

   ii. Use ossfs. Add the -oram_role option.

   The bucket named Bucket1 in the China (Hangzhou) region is used in the example:

```
ossfs bucket1 /tmp/ossfs -ourl=http://oss-cn-hangzhou.aliyuncs.com -oram_role=http:
//100.100.100.200/latest/meta-data/ram/security-credentials/EcsRamRoleOssTest
```

Configure access permissions

By default, the directory to which ossfs attaches files can be accessed only by the owner (the user who performs the mount operation) of the mount point. To modify the default permission settings to allow other users or user groups to access the mount point, you can use the following options while you run ossfs:

- allow_other: authorizes other users to access the directory to which the bucket is attached, but not objects in the directory. To modify the access permission on the objects in the directory, you must run the chmod command. No value is available for this option. To grant permissions to other users, use the -oallow_other option.
- uid: specifies the user ID (UID) of the owner of a directory.
- gid: specifies the group ID (GID) of the owner of a directory.
- mp_umask: specifies the permission mask set for the mount point. This option takes effect only when the allow_other option is set. Default value: 000. This option is used in the same way as the **umask** command. For example, you can use the -oallow_other -omp_umask=007 option to set the permission of the mount point to 770, and use the -oallow_other -omp_umask=077 option to set the permission of the mount point to 700.

Configuration examples:

- Set the permission to 777 to allow access from all users.

```
ossfs bucket_name mount_point -ourl=endpoint -oallow_other
```

- Set the permission to 770 to allow access from users only in the same group as the owner of the mount point.

```
ossfs bucket_name mount_point -ourl=endpoint -oallow_other -omp_umask=007
```

- When you attach the bucket, specify the user and the user group, and then set the permission to 770 to allow access from users only in the same group.

  The user www is used in the example. You can run the **id** command to obtain the UID or GID of the user, and then specify the uid or gid parameter when you attach the bucket.

```
id www
uid=1000(www) gid=1000(web) groups=1000(web)
ossfs bucket_name mount_point -ourl=endpoint -oallow_other -ouid=1000 -ogid=1000 -omp_uma
sk=007
```

Attach a specific directory

You can use ossfs to attach a specific directory in a bucket to the local file system by specifying a prefix. Command syntax:

```
ossfs bucket:/prefix mount_point -ourl=endpoint
```

When you run this command, make sure that an object named *${prefix}/* exists in the bucket. You can run the stat command of ossutil to check whether the object exists.

The following command provides an example on how to attach the *folder* directory in the bucket-ossfs-test bucket in the China (Hangzhou) region to */tmp/ossfs-folder*.

```
ossfs bucket-ossfs-test:/folder /tmp/ossfs-folder -ourl=http://oss-cn-hangzhou.aliyuncs.com
```

Attach a directory on startup

1. Write the bucket name, AccessKey ID, and AccessKey secret to the */etc/passwd-ossfs* file, and change the permission on the file to 640.

   For more information, see Installation.

2. Enable automatic mount on startup.

   The method to enable automatic mount on startup varies with the version of your operating system.

   - Enable automatic mount on startup by using the fstab file for Ubuntu 14.04 and CentOS 6.5

     a. Add the following command to the */etc/fstab* file:

```
ossfs#bucket_name mount_point fuse _netdev,url=url,allow_other 0 0
```

     b. Save the */etc/fstab* file. Run the **mount -a** command. The settings are correct if no errors are reported.

     c. After you complete the preceding operations, automatic mount on startup is enabled in Ubuntu 14.04. For CentOS 6.5, you must also run the following command:

```
chkconfig netfs on
```

   - Enable automatic mount on startup by using the script for CentOS 7.0 or later

a. Create the *ossfs* file in the */etc/init.d/* directory, and copy the content of the template to this file. Replace your_xxx with your actual information.

b. Run the following command to allow the *ossfs* script to be executed:

```
chmod a+x /etc/init.d/ossfs
```

After the preceding command is run, you can execute the script. If the content of the script is correct, the OSS bucket is attached to the specified directory.

c. Run the following command to start the *ossfs* script as a service which is automatically enabled on startup:

```
chkconfig ossfs on
```

d. After you complete the preceding operations, automatic mount on startup is enabled for ossfs.

Start ossfs by using Supervisor

Supervisor is a universal process management program of Python. Supervisor can turn a general command-line process into a background daemon and monitor the process. Supervisor automatically restarts the process when the process stops unexpectedly. Perform the following steps to start ossfs by using Supervisor:

1. Install Supervisor.
   - CentOS

     ```
     yum install supervisor
     ```

   - Ubuntu

     ```
     sudo apt-get install supervisor
     ```

2. Create an ossfs startup script.

   i. Create the *start_ossfs.sh* file.

   ```
   mkdir /root/ossfs_scripts
   vi /root/ossfs_scripts/start_ossfs.sh
   ```

   ii. Write the startup script.

   ```
   # Remove the mount point.
   fusermount -u /mnt/ossfs
   # Attach the OSS bucket again. You must use the -f parameter to run ossfs on the fr
   ontend.
   exec ossfs bucket_name mount_point -ourl=endpoint -f
   ```

3. Edit the */etc/supervisor/supervisord.conf* file. Add the following content to the end of the file:

```
[program:ossfs]
command=bash /root/ossfs_scripts/start_ossfs.sh
logfile=/var/log/ossfs.log
log_stdout=true
log_stderr=true
logfile_maxbytes=1MB
logfile_backups=10
```

4. Run Supervisor.

```
supervisord
```

5. Verify whether Supervisor runs properly.

```
ps aux | grep supervisor # The Supervisor process is displayed.
ps aux | grep ossfs # The ossfs process is displayed.
kill -9 ossfs # Terminate the ossfs process. Supervisor restarts ossfs. Do not run the killall command because this command sends the SIGTERM signal to terminate the ossfs process. In this case, Supervisor will not restart ossfs.
ps aux | grep ossfs # The ossfs process is displayed.
```

Enable debug logging

You may encounter issues when you use ossfs. In this case, you must enable the debug logging feature to analyze and locate the issues based on the logs. You can enable debug logging by using one of the following methods:

- Add the -d -odbglevel=debug -ocurldbg option when you attach a bucket to a local directory. ossfs writes the logs to the system logs.
  - CentOS

    Logs are stored in */var/log/messages*.
  - Ubuntu

    Logs are stored in */var/log/syslog*.
- Add the -d -odbglevel=debug -ocurldbg -f option when you attach a directory. ossfs displays the logs.

5.4. Common options

You can use the -h option to view the common options supported by ossfs.

Command syntax

```
./ossfs -h
```

You must use this command in the directory where *ossfs* is located. The directory is */usr/local/bin/* by default and is subject to the actual installation environment.

Common options

ossfs is implemented based on Filesystem in Userspace (FUSE) and supports the fuse options in addition to the ossfs options. When you attach a bucket, you can set different startup options. The options can be in one of the following formats:

```
-o option_name[=option_value]  or  -ooption_name[=option_value]
```

For example, you can use the following option to specify the uid and gid parameters when you attach a bucket:

```
ossfs bucket_name mount_point -ourl=endpoint -ouid=uid -ogid=gid
```

The following section describes common ossfs options.

- url: specifies the endpoint used to access a bucket. Format: `url=endpoint`. The default request protocol is HTTP.

  Example:

  ```
  -ourl=oss-cn-hangzhou.aliyuncs.com
  -ourl=http://oss-cn-hangzhou.aliyuncs.com
  -ourl=https://oss-cn-hangzhou.aliyuncs.com
  ```

- passwd_file: specifies the object that stores the AccessKey pair used to access a bucket. Default value: */etc/passwd-ossfs*. Make sure that the permission of this object is configured correctly. If the object is */etc/passwd-ossfs*, you can set the permission to 640. If the object is not */etc/passwd-ossfs*, you must set the permission to 600. The content of the object is in the following format: `${bucket}:${access-key-id}:{access-key-secret}`.

  Example:

  ```
  echo bucket-test:LTAIbZcdVCmQ****:MOk8x0y9hxQ31coh7A5e2MZEUz**** > /etc/passwd-ossfs
  chmod 640 /etc/passwd-ossfs
  echo bucket-test:LTAIbZcdVCmQ****:MOk8x0y9hxQ31coh7A5e2MZEUz**** > /passwd-path/passwd-ossfs
  chmod 600 /passwd-path/passwd-ossfs
  -opasswd_file=/passwd-path/passwd-ossfs
  ```

- max_stat_cache_size: specifies the maximum number of objects whose metadata can be cached. Default value: 1000. When a directory contains a large number of objects, you can adjust this option to accelerate object listing when you run the ls command to list objects. To disable metadata caching, you can set the value of this option to 0.
- allow_other: authorizes other users to access the directory to which the bucket is attached, but not objects in the directory. To modify the access permission on the objects in the directory, you must run the chmod command. No value is available for this option. To grant permissions to other users, use the -oallow_other option.
- dbglevel: specifies the log level. Valid values: critical, error, warn, info, and debug. Default value: critical. For example, to enable info log collection, use the -odbglevel=info option. Logs are written to system logs. For example, in CentOS, logs are written to */var/log/messages*.
- -f: runs ossfs as a foreground program instead of a daemon. In this case, logs are displayed on the screen of a terminal. This option is used in debugging.
- -d: enables logging. This option is also used in fuse. In ossfs, this option is equivalent to the -odbglevel=info setting.

Options

Unless otherwise specified, options are in one of the following formats: `-ooption_name=option_value` or `-o option_name=option_value`.

- ossfs options

Option	Description

Option	Description
default_acl	Specifies the access control list (ACL) of an object when the object is uploaded to OSS. Default value: private. Valid values: ◦ *private* ◦ *public-read* ◦ *public-read-write* For more information about ACLs, see ACL.
retries	Specifies the number of retry attempts when a request fails. Default value: 2.
storage_class	Specifies the storage class of an object when the object is uploaded to OSS. Default value: Standard. Valid values: ◦ *Standard* ◦ *IA* ◦ *Archive* For more information about storage classes, see Overview.
public_bucket	Allows users to access buckets as anonymous users. This option is applicable only to buckets whose ACLs are public read/write. Default value: 0. Valid values: ◦ *0*: prohibits users to access buckets as anonymous users. ◦ *1*: allows users to access buckets as anonymous users.
passwd_file	Specifies the object that stores the AccessKey pair used to access a bucket. Default value: */etc/passwd-ossfs*.
connect_timeout	Specifies the timeout period in seconds for connections. Default value: 300.
readwrite_timeout	Specifies the timeout period in seconds for read or write requests. Default value: 60.
max_stat_cache_size	Specifies the maximum number of objects whose metadata can be cached. Default value: 1000. The metadata of 1,000 objects consumes about 4 MB of cache space.
stat_cache_expire	Specifies the expiration time for the object metadata cache. By default, the metadata cache does not expire.
no_check_certificate	Specifies that the server certificates are not validated. This option is valid only when the request protocol is HTTPS. By default, certificate validation is enabled. No values are available for this option. To disable certificate validation, use the -ono_check_certificate option.
multireq_max	Specifies the maximum number of concurrent requests to access object metadata during object listing. Default value: 20.
parallel_count	Specifies the number of parts that can be uploaded concurrently when multipart upload is used to upload large objects. Default value: 5.

Option	Description
multipart_size	Specifies the size of each part in MB when multipart upload is used to upload data. Default value: 10. This option limits the maximum size of the object to upload. When multipart upload is used, the maximum number of parts that can be uploaded is 10,000. By default, the maximum size of the object that can be uploaded is 100 GB. You can adjust the value of this option to upload larger objects.
url	Specifies the endpoint used to access a bucket.
mp_umask	Specifies the permission mask for mount points. This option takes effect only when the allow_other option is set. Default value: 000. This option is used in the same way as the **umask** command. For example, you can set -oallow_other -omp_umask=007 to set the permission of the mount point to 770, and set -oallow_other -omp_umask=077 to set the permission of the mount point to 700.
enable_content_md5	Specifies whether to set the CONTENT_MD5 header for uploads. By default, this header is not set. No value is available for this option. To specify CONTENT_MD5, use the -oenable_content_md5 option.
ram_role	Specifies access to OSS by using instance RAM roles. When you access OSS by using instance RAM roles, the AccessKey ID and AccessKey secret of the key object are ignored.
dbglevel	Specifies the log level. Valid values: critical, error, warn, info, and debug. Default value: critical.
curldbg	Specifies whether to enable libcurl logging. By default, libcurl logging is disabled. No value is available for this option. To obtain libcurl logs, use the -ocurldbg option.

- fuse options

Option	Description
allow_other	Modifies the permission of the mount point to allow access from all users. By default, only the root user can set this option. No value is available for this option. To allow access from all users, use the -oallow_other option.
uid	Specifies the user ID (UID) of the owner of a directory.
gid	Specifies the group ID (GID) of the owner of a directory.

5.5. FAQ

This topic describes possible problems that you may encounter when you use ossfs and their solutions.

ossfs errors are reported with messages. You must collect these messages to determine what problems have occurred and how to troubleshoot them. For examples, you may encounter socket connection failures or errors with HTTP status codes 4xx and 5xx. You must enable the debug logging feature before troubleshooting.

- Errors with HTTP status code 403 occur when access is denied because the user is unauthorized.
- Errors with HTTP status code 400 occur due to incorrect operations.
- Errors with HTTP status code 5xx occur due to network jitters and client services.

> **Note** If ossfs cannot meet your business requirements, you can use ossutil.
>
> - ossfs mounts remote OSS buckets to local disks. We recommend that you do not use ossfs to handle services that require high read and write performance.
> - ossfs operations are not atomic, so there is a risk that local operations may succeed while OSS remote operations fail.

Case: The "conflicts with file from package fuse-devel" message is returned when you use yum/apt-get to install ossfs.

Analysis: This error occurs because an earlier version of fuse is present in the system and conflicts with the dependent version of ossfs.

Solution: Use a package manager to uninstall fuse and then reinstall ossfs.

Case: The "fuse: warning: library too old, some operations may not work" message is returned during the installation of ossfs.

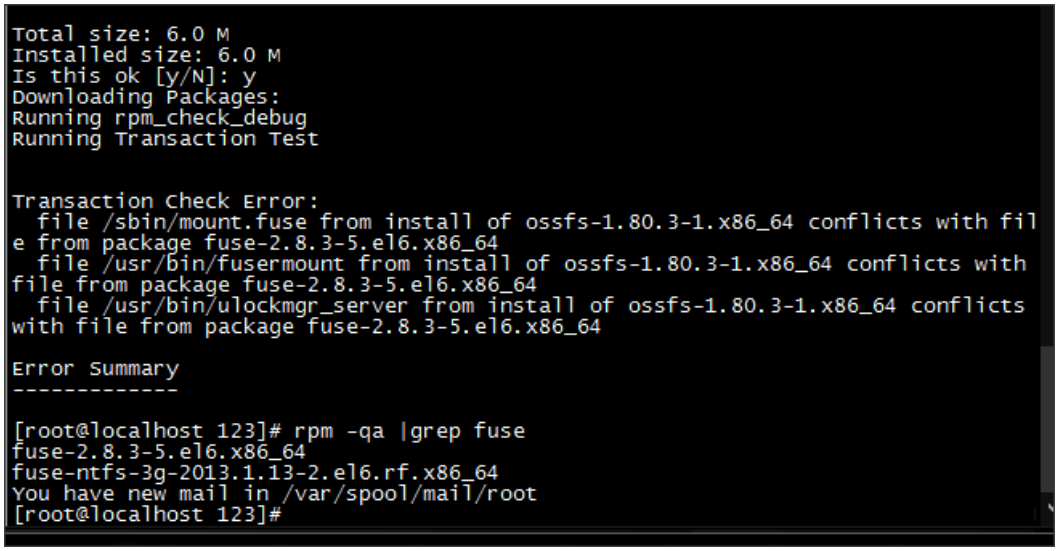
Analysis: This error usually occurs because you install libfuse on your own, and the libfuse version used during ossfs compilation is later than that is linked to ossfs during runtime. The ossfs installation package provided by Alibaba Cloud contains libfuse 2.8.4. When you install ossfs in CentOS 5.x or CentOS 6.x, if libfuse 2.8.3 already exists in the system and is linked to ossfs, this error will occur.

You can use the **ldd $(which ossfs) | grep fuse** command to check the fuse version that is linked to ossfs during runtime. If the command output is */lib64/libfuse.so.2*, you can use the **ls -l /lib64/libfuse*** command to check the fuse version.

Solution: Link ossfs to the fuse version provided in the ossfs installation package.

1. Run the **rpm -ql ossfs | grep fuse** command to locate the directory of libfuse.
2. If the command output is */usr/lib/libfuse.so.2*, use the **LD_LIBRARY_PATH=/usr/lib ossfs...** command to run ossfs.

Case: An error occurs during the installation of the fuse dependency library.



Analysis: This error occurs because the version of fuse does not meet the requirements of ossfs.

Solution: Download and install the latest version of fuse. Do not use yum to install fuse. For more information, visit libfuse.

Case: The "oss: unable to access MOUNTPOINT /tmp/ossfs: Transport endpoint is not connected" message is returned while ossfs mounts an OSS bucket.

Analysis: This error occurs because the destination directory of the OSS bucket is not created.

Solution: Create the destination directory, and then mount the bucket.

Case: The "fusermount: failed to open current directory: Permission denied" message is returned while ossfs mounts an OSS bucket.

Analysis: This error occurs due to a bug in fuse that requires you to have the read permissions on the current directory instead of the destination directory of the OSS bucket.

Solution: Run the cd command to switch to a directory on which you have the read permissions, and then use ossfs to mount the bucket.

Case: The "ossfs: MOUNTPOINT directory /tmp/ossfs is not empty. if you are sure this is safe, can use the 'nonempty' mount option" message is returned while ossfs mounts an OSS bucket.

Analysis: By default, ossfs can only mount an OSS bucket to an empty directory. This error occurs when ossfs attempts to mount a bucket to a directory that is not empty.

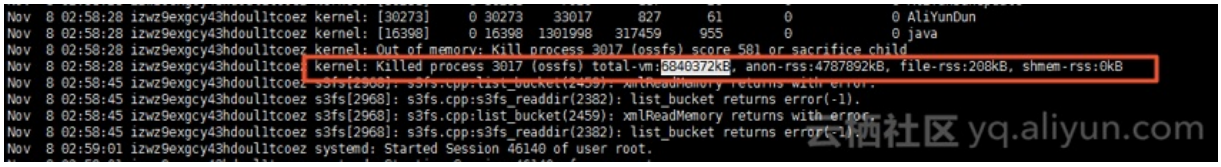
Solution: Switch to an empty directory and re-mount the bucket. If you still want the bucket to be mounted to the current directory, you must use the -ononempty option.

Case: The "operation not permitted" message is returned when the ls command is run to list objects in the directory after a bucket is mounted.

Analysis: The file system has strict limitations on object names and directory names. This error occurs when the names of objects in your buckets contain non-printable characters.

Solution: Use a tool to rename the objects, and then run the ls command. Objects in the directory are displayed.

Case: ossfs is disconnected occasionally.



Analysis:

1. ossfs debug logs are collected and the -d -o f2 parameter is used. ossfs logs are written in the `/var/log/message` file of the system.
2. Log analysis shows that ossfs requests too much memory for the listbucket and listobject operations, which triggers an Out Of Memory (OOM) error.

> **Note** The listobject operation sends an HTTP request to OSS to obtain object metadata. If you have a large number of objects, running the ls command requires a large amount of memory to obtain the object metadata.

Solution:

- Specify the -omax_stat_cache_size=xxx parameter to increase the size of stat cache. In this case, the first running of the ls command will be slow, but subsequent running of the command will be fast because metadata of the objects is stored in the local cache. The default value of this parameter is 1000. The metadata of 1,000 objects consumes about 4 MB of memory. You can adjust the value based on the memory size of your machine.
- ossfs writes a large number of files in TempCache during read or write operations, which is similar to NGINX. this may result in insufficient disk space. Therefore, you can frequently clear cache space to solve the problem.
- Use ossutil instead of ossfs. You can use ossfs for services that do not require high real-timeliness. We recommend that you use ossutil for services that require high reliability and stability.

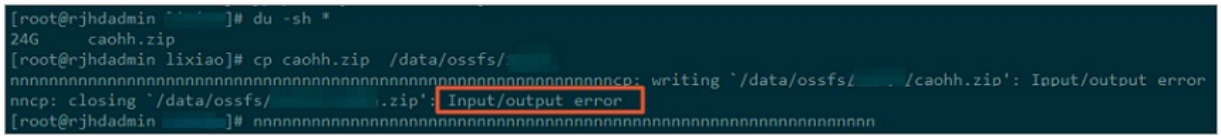
Case: The "THE bucket you are attempting to access must be addressed using the specified endpoint" message is returned when you access a bucket.

Analysis: This error occurs because you are not using the correct endpoint to access the bucket. This error may occur in the following scenarios.

- The bucket and endpoint do not match.
- The UID of the bucket owner is different from that of the Alibaba Cloud account that corresponds to the AccessKey pair.

Solution: Check whether the configurations are correct and modify the configurations if necessary.

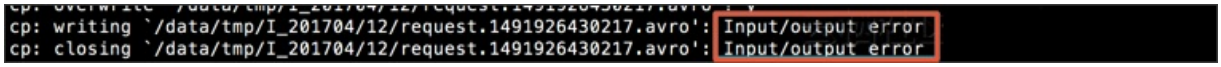
Case: The "input/output error" message is returned when the cp command is used to copy data.



Analysis: This error occurs when system disk errors are captured. You can check if there are heavy read and write loads on the disk when this error occurs.

Solution: Specify multipart parameters to control the speed of read and write operations on objects. You can use the **ossfs -h** command to view the multipart parameters.

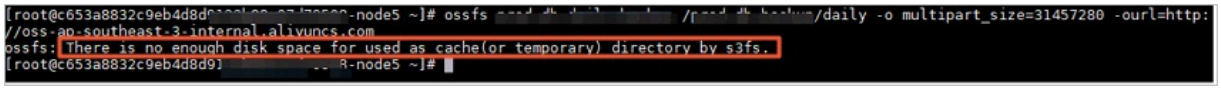
Case: The "input/output error" message is returned during synchronization by using rsync.



Analysis: This error occurs when ossfs is used with rsync. In this case, the cp command is used to copy a large object of 141 GB in size, causing very heavy read and write loads on the disk.

Solution: Use ossutil to download OSS objects to a local ECS instance or upload objects from a local device to an ECS instance in multipart mode.

Case: The "There is no enough disk space for used as cache(or temporary)" message is returned when ossfs is used to upload a large object.



Analysis: ossfs uploads a large object in multipart mode. The default size of each part is 10 MB. The maximum number of parts that can be uploaded is 1,000.

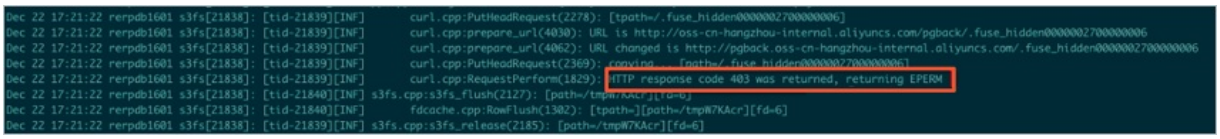
When ossfs uploads an object, it writes temporary cache files to the */tmp* directory. The available space of the disk in which the */tmp* directory is located must be greater than the total size of the object to upload. If the available disk space is smaller than the total size of the object to upload, this error will occur. This error may occur in the following scenarios:

- Scenario 1: The available disk space is smaller than the total size of the object to upload. For example, the available disk space is 200 GB, but the size of the object to upload is 300 GB.
- Scenario 2: The parameters for the part size and the number of upload threads are invalid. For example, the available disk space is 300 GB, and the size of the object to upload is 100 GB. The multipart_size parameter is set to 100 GB, and the number of upload threads is set to 5. In this case, ossfs determines that the size of the object to upload is 100 GB × 5 = 500 GB, which is greater than the available space of the disk.

Solution:

- Scenario 1: Increase the available space of the disk.
- Scenario 2: Set the part size in MB and the maximum number of parts to 1,000.

Case: A 403 error occurs when the touch command is run on an object in the mounted bucket.



Analysis: This error usually occurs when the operation is unauthorized. Possible scenarios are as follows:

- The object is of the Archive storage class.
- You are not authorized to manage the bucket by using your AccessKey pair.

Solution:

- Restore the archived object to access it.
- Grant necessary permissions to the account that uses the AccessKey pair.

Case: The values of the Content-Type parameters of all the objects that are uploaded to OSS by using ossfs are application/octet-stream.

Analysis: When uploading an object, ossfs queries the */etc/mime.types* file to set the Content-Type parameter for the object. If the mime.types file does not exist, the value of the Content-Type parameter is set to application/octet-stream.

Solution: Check whether the mime.types file exists. If the file does not exist, add it.

- Add the *mime.types* file by using commands
  - Ubuntu

    Use the **sudo apt-get install mime-support** command to add the file.
  - CentOS

    Use the **sudo yum install mailcap** command to add the file.
- Manually add the *mime.types* file
  i. Create the *mime.types* file.

  ```
  vi /etc/mime.types
  ```

  ii. Add the desired type in `application/javascript js` format, with one line per type.

Mount OSS again.

Why is running the ls command very slow when the directory contains many objects?

Analysis: If a directory contains N objects, OSS HTTP requests must be initiated N times for the ls command to be run to list the N objects in the directory. This can cause serious performance problems when the number of objects is large.

Solution: Specify the `-omax_stat_cache_size=xxx` parameter to increase the size of stat cache. In this case, the first running of the ls command will be slow, but subsequent running of the command will be fast because metadata of the objects is stored in the local cache. The default value of this parameter is 1000. The metadata of 1,000 objects consumes about 4 MB of memory. You can adjust the value based on the memory size of your machine.

Why is the information such as the size of an object displayed different from that displayed when other tools are used?

Analysis: When ossfs mounts a bucket, object metadata such as the size and ACL information of the objects in the bucket is cached. This can help facilitate the running of the ls command because it saves sending a request to OSS each time when the ls command is run. However, if the user modifies the object by using SDKs, the OSS console, or ossutil but ossfs does not obtain the updated object metadata in time, the information displayed will be different.

Solution: Specify the -omax_stat_cache_size=0 parameter to disable the metadata caching feature. In this case, a request is sent to OSS to obtain the latest object metadata each time when the ls command is run.

What do I do to avoid the cost of scanning objects by background programs when ossfs mounts OSS buckets to ECS instances?

Analysis: When a program scans a directory in which ossfs mounts a bucket, a request is sent to OSS. If many requests are sent, fees will be incurred.

Solution: Use the auditd tool to check which program scans the directory in which ossfs mounts the bucket. Follow these steps:

1. Install and start auditd.

   ```
   sudo apt-get install auditd
   sudo service auditd start
   ```

2. Set the directory to monitor. For example, use the following command to monitor the */mnt/ossfs* directory.

   ```
   auditctl -w /mnt/ossfs
   ```

3. Check the audit log to view which programs have accessed the directory.

   ```
   ausearch -i | grep /mnt/ossfs
   ```

4. Set parameters to skip scheduled program scans.

   For example, if you find that the updatedb program scans the directory, you can modify the */etc/updatedb.conf* configuration file to skip scanning by this program. Follow these steps:

   i. Set `RUNEFS=` to `fuse.ossfs`.

   ii. Set `PRUNEPATHS=` to the directory.

6.ossftp

6.1. Overview

ossftp is an FTP server tool based on Alibaba Cloud Object Storage Service (OSS). ossftp maps operations related to files and directories to those on OSS objects and directories. This way, you can manage objects stored in OSS over FTP.

Usage notes

- ossftp is provided for individual tests. To manage your OSS resources, we recommend that you use tools such as the OSS console, ossutil, ossbrowser, and SDK in production environments.
- The FTP protocol transmits data in plaintext. To prevent password leaks, we recommend that you run ossftp and the client on the same machine and access data by using `127.0.0.1:port`.

Deployment environments

- Supported operating systems: Windows, Linux, and macOS
- Supported architectures: x86 (32-bit and 64-bit)
- Runtime environments: Python2.7 and 3.x

Features

- Upload, download, and delete objects and directories.
- Use multipart upload to upload large objects.
- Support most FTP commands.

6.2. Installation

You can run ossftp on operating systems such as Windows, Linux, and macOS. This topic describes how to install and use ossftp.

Download URLs

You can download the installation package of ossftp based on your system environment:

- Windows: ossftp-1.2.0-win.zip

  By default, Python 2.7 is not installed on Windows. The installation package includes Python 2.7. You can run ossftp directly after the package is decompressed.

- Linux and macOS: ossftp-1.2.0-linux-mac.zip

  By default, Python 2.7 or Python 3.x is installed on Linux or macOS. The installation package contains only the required dependent libraries.

Procedure

1. Decompress the downloaded installation package.

   The path to which the installation package is decompressed cannot contain Chinese characters.

2. Run ossftp.

   After ossftp is run, TCP ports 2048 and 8192 of the local computer are enabled by default. Port

2048 is used as the port of the FTP service to receive FTP requests. Port 8192 is used as the port of the web service to open the graphical management interface of ossftp. To provide services for other users, enable the two ports in firewall configurations. The method to run ossftp varies with operating systems.

- In Windows:

  Decompress the installation package and double-click *start.vbs*. If no response is returned, upgrade the installed version of Internet Explorer, or set the default browser to a different browser.

- In Linux:

  a. Run the following command to decompress the downloaded file:

  ```
  unzip ossftp-1.0.3-linux-mac.zip
  ```

  b. Run the following commands to open the decompressed folder and run *start.sh*:

  ```
  cd ossftp-1.0.3-linux-mac
  bash start.sh
  ```

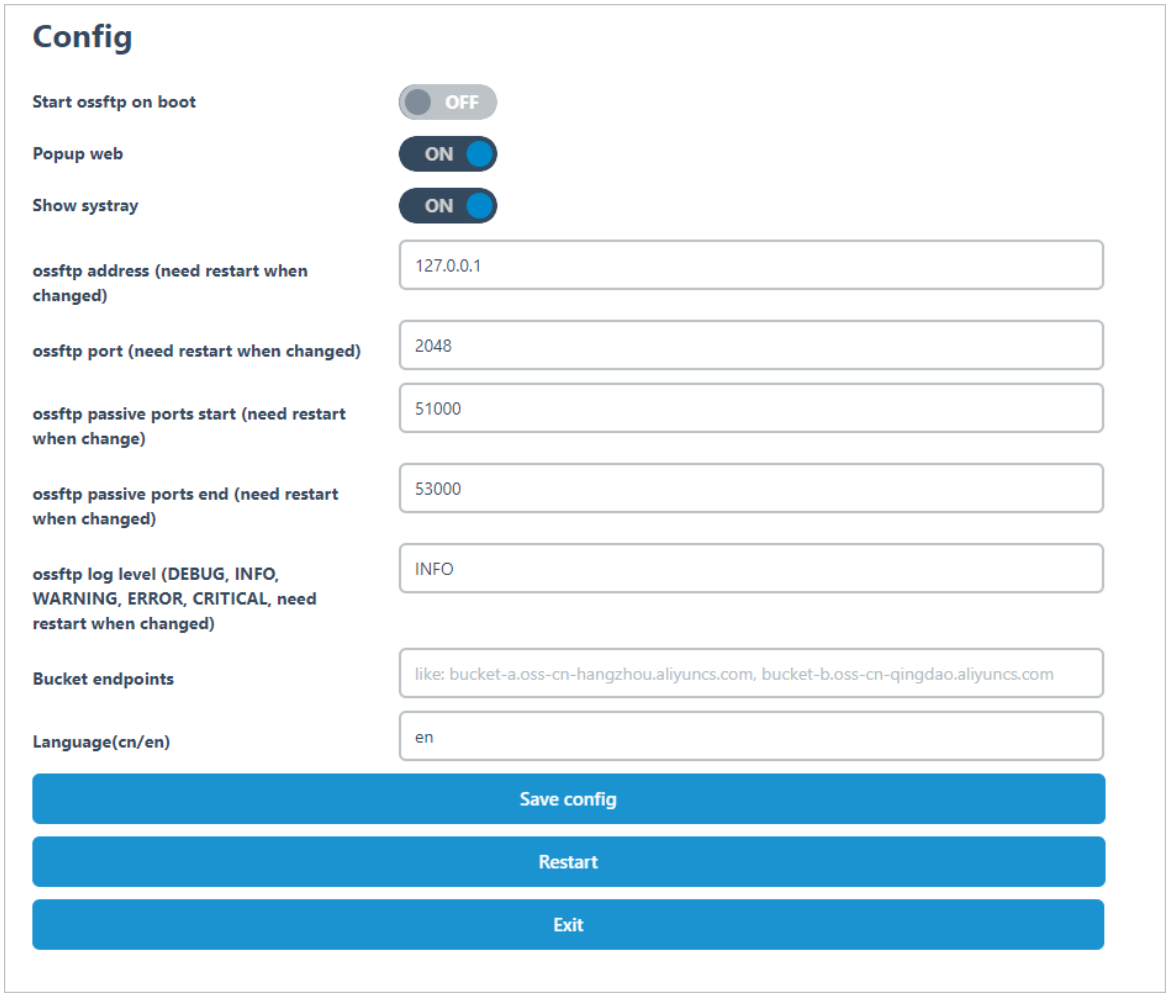
  c. Use a browser to access the graphical management interface of ossftp. The endpoint is `http://127.0.0.1:8192`.

  If the local machine does not provide a graphical management interface, you can use other computers to access the graphical management interface of ossftp. The endpoint is in the `http://Linux server IP address:8192` format.

- In macOS:

  Decompress the installation package and double-click *start.command*. You can also run **$ bash start.command** on the command line.

3. The following table describes the parameters that you can configure on the graphical management interface of ossftp.



Parameter	Description
ossftp address	Enter the IP address of the client that needs to use the FTP service. If the client is running on the local computer, you can keep the default settings.
ossftp port	Set the port that receives requests for ossftp. If the default port does not conflict with other ports, keep the default port.
ossftp passive ports start	Set the starting port number of the range to respond to requests for ossftp. If the default port does not conflict with other ports, keep the default port.
ossftp passive ports end	Set the ending port number of the range to respond to requests for ossftp. If the default port does not conflict with other ports, keep the default port.

Parameter	Description
ossftp log level	Set the log output level of ossftp. Valid values: ○ DEBUG: ossftp records information events in detail. DEBUG logs are used to debug programs. ○ INFO: ossftp records events that occur when software properly runs. ○ WARNING: ossftp records abnormal events that do not affect the system. ○ ERROR: ossftp records abnormal events that affect the system but do not affect the system reliability. ○ CRITICAL: ossftp records events that cause the system to fail to work.
Bucket endpoints	Enter the endpoint of the bucket. The format is `BucketName.Endpoint`. Separate multiple endpoints with commas (,). Example: `examplebucket.oss-cn-hangzhou.aliyuncs.com`.
Language	Select a display language for ossftp.

4. After you configure the parameters, click **Save config**. Then, click **Restart**.

   Do not click **Exit**. Otherwise, ossftp stops running.

5. Install the FTP client.

   The FileZilla software is used in the example. For more information about the address to download the software, visit FileZilla.

6. Open FileZilla. Configure the OSS access information. Then, click **Quickconnect**.

Parameter	Description
Host	Configure the IP addresses of the server. If the server and client reside on the same device, use default address 127.0.0.1.

Parameter	Description
Username and **Password**	Enter the username and password to connect to ossftp. You can use the following methods to obtain the username and password: ○ AccessKey ■ When you use an AccessKey pair to connect to ossftp, the username consists of a bucket name and an AccessKey ID that has the access permissions on the bucket. The format is `AccessKey ID/Bucket name`. Example: `Y6IoUOZReouXvWaXuwjvDch9******/examplebucket`. ■ The password is the AccessKey secret that is paired with the AccessKey ID. For more information about how to obtain an AccessKey pair, see Obtain an AccessKey pair. ○ Custom username You can generate a custom username on the ossftp server for the client. For more information about configuration methods, see Appendix: Create a custom logon account.
Port	Enter the listening port configured for ossftp.

Note The ossftp server can be connected to only one client at a time. Subsequent connection requests cause the existing connection to disconnect from the client.

7. Upload and download objects.

   In the FileZilla console, drag an object from the local site on the left side to the remote site on the right side. This way, the object is uploaded. To download an object, drag the object from the remote site to the local site.

Appendix: Create a custom logon account

Open the *config.json* file in the installation directory of ossftp. Modify parameters in `accounts`. Keep default settings for other parameters. Configuration example:

```
{
  "modules":{
    "accounts":[
      {
        // Specify the AccessKey secret and the AccessKey ID that has the permissions to ac
cess the required bucket.
        "access_id":"LTAI4FrfJPUSoKm4JH******",
        "access_secret":"Y6IoUOZReouXvWaXuwjvDch9******",
        // Specify the name of the bucket.
        "bucket_name":"examplebucket",
        // Specify the URLs of the objects in the bucket. After the preceding information i
s specified, the account can access the objects whose URLs are specified. If no URLs are sp
ecified, all objects in the bucket can be accessed.
        "home_dir":"examplefolder/",
        // Customize a password for logons.
        "login_password":"password1",
        // Customize a username for logons.
        "login_username":"user1"
      },
      {
        "access_id":"LTAI4FrfJPUSoKm4JH******",
        "access_secret":"Y6IoUOZReouXvWaXuwjvDch9******",
        "bucket_name":"examplebucket",
        "home_dir":"",
        "login_password":"password2",
        "login_username":"user2"
      }
    ],
    "launcher":{
      "auto_start":0,
      "control_port":8192,
      "language":"cn",
      "popup_webui":1,
      "show_systray":1
    },
    "ossftp":{
      "address":"127.0.0.1",
      "bucket_endpoints":"",
      "log_level":"INFO",
      "passive_ports_start":51000,
      "passive_ports_end":53000,
      "port":2048
    }
  }
}
```

After the configuration file is saved, you must restart the ossftp service on the graphical management interface of ossftp. Otherwise, the configured account does not take effect.

6.3. Use cases

6.3.1. How to store remote attachments from Discuz! to OSS

This topic describes how to store remote attachments from Discuz! to OSS.

Prerequisites

- OSS is activated and a bucket whose ACL is Public Read is created.
  - For more information about how to activate OSS, see Activate OSS.
  - For more information about how to create a bucket, see Create buckets.
- A Discuz! forum is built.

Context

The website remote attachment function allows you to directly store uploaded attachments to a remote storage server, which is usually a remote FTP server. Currently, Discuz! forums, PHPWind forums, and WordPress websites support the remote attachment function.

The version of Discuz! used for this topic is Discuz! X3.1.

Procedure

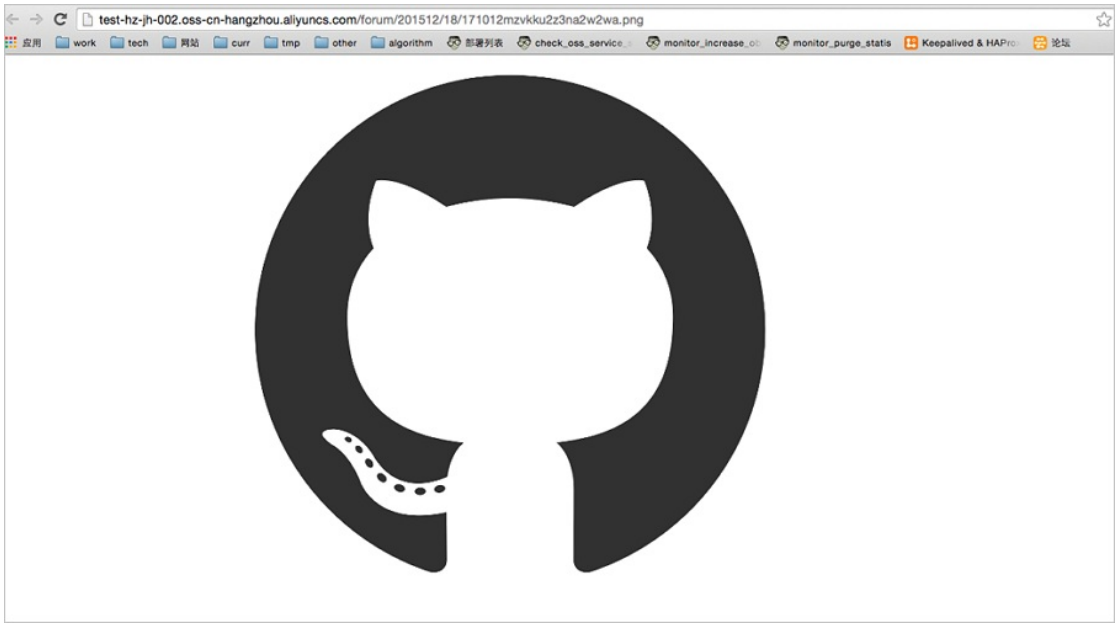
1. Log on to the Discuz! website with an administrator account.
2. On the management page, choose **Global > Upload Settings**.
3. Click the **Remote Attachment** tab and configure parameters.

Configuration item	Description
Enable remote attachment	Specifies whether to enable remote attachment. Select **Yes**.
Enable SSL connection	Specifies whether to enable SSL connection. Select **No**.
FTP server	Specifies the IP address that runs ossftp. We recommend that you set this parameter to 127.0.0.1.
FTP server port number	Specifies the port number of the FTP server. The default value is 2048.
FTP account	Specifies the FTP account in AccessKeyID/BucketName format. Note that the forward slash (/) is a delimiter and does not indicate an alternative.
FTP password	Specifies the FTP password AccessKey secret.
Passive mode connection	Specifies whether to enable passive mode connection. Select **Yes**.

Configuration item	Description
Remote upload directory	Specifies the remote upload directory for attachments. We recommend that you set this parameter to a period (.) to create a directory for attachments in the root directory of the bucket.
Remote access URL	Specifies the public endpoint of the bucket. The format is http://BucketName.Endpoint. In this example, the bucket resides in the China (Hangzhou) region, and the bucket name is test-hz-jh-002. Therefore, the URL is http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com. For more information about endpoints, see OSS domain names.
FTP timeout (seconds)	Specifies the FTP timeout period. Set the value to 0, indicating that the default setting of the server applies.

4. After the configuration is complete, click **Test Remote Attachment** to verify whether the configuration is correct.
5. Post a new article to verify whether the configuration is successful.
   i. Upload an image attachment to the article.
   ii. Right-click the image and choose **Open Link in New Tab** from the shortcut menu.

   The URL in the following figure indicates that the image has been uploaded to the OSS bucket test-hz-jh-002.



6.3.2. How to store remote attachments from PHPWind to OSS

This topic describes how to store remote attachments from PHPWind to OSS.

Prerequisites

- OSS is activated and a bucket whose ACL is Public Read is created.
  - For more information about how to activate OSS, see Activate OSS.
  - For more information about how to create a bucket, see Create buckets.
- A PHPWind forum is built.

Context

The website remote attachment function allows you to directly store uploaded attachments to a remote storage server, which is usually a remote FTP server. Currently, Discuz! forums, PHPWind forums, and WordPress websites support the remote attachment function.

PHPWind8.7 is used for this topic.

Procedure

1. Log on to the PHPWind website with an administrator account.
2. On the management page, choose **Global > Upload Settings > Remote Attachments**.
3. Click the **FTP Settings** tab and configure parameters.

Configuration item	Description
Enable FTP uploads	Specifies whether to enable FTP uploads. Select **Enable**.
Website attachment URL	Specifies the public endpoint of the bucket. The format is http://BucketName.Endpoint. In this example, the bucket resides in the China (Hangzhou) region, and the bucket name is test-hz-jh-002. Therefore, the URL is http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com. For more information about endpoints, see OSS domain names.
FTP server	Specifies the IP address that runs ossftp. We recommend that you set this parameter to 127.0.0.1.
FTP server port number	Specifies the port number of the FTP server. The default value is 2048.
Remote upload directory	Specifies the remote upload directory for attachments. We recommend that you set this parameter to a period (.) to create a directory for attachments in the root directory of the bucket.
FTP account	Specifies the FTP account in AccessKeyID/BucketName format. Note that the forward slash (/) is a delimiter and does not indicate an alternative.
FTP password	Specifies the FTP password AccessKey secret.
FTP timeout (seconds)	Specifies the FTP timeout period. Set the value to 10. If no result is returned within 10 seconds, the system returns a timeout response.

4. Post a new article to verify whether the configuration is successful.
   i. Upload an image attachment to the article.

ii. Right-click the image and choose **Open Link in New Tab** from the shortcut menu.

6.3.3. How to store remote attachments from WordPress websites to OSS

This topic describes how to store remote attachments from WordPress websites to OSS.

Prerequisites

- OSS is activated and a bucket whose ACL is public read is created.
  - For more information about how to activate OSS, see Activate OSS.
  - For more information about how to create a bucket, see Create buckets.
- A WordPress website is built.

Context

The website remote attachment feature allows you to directly store uploaded attachments to a remote storage server such as a remote FTP server. Discuz! forums, PHPWind forums, and WordPress websites support the remote attachment feature.

WordPress does not provide native support for this feature, but you can use a third-party plug-in to implement this feature. In this topic, the WordPress version is 4.3.1 and the plug-in is Hacklog Remote Attachment.

Procedure

1. Log on to the WordPress website by using an administrator account.
2. Click **Plug-in**. Enter **FTP** in the *Keyword* search bar. Press Enter.
3. Find Hacklog Remote Attachment. Click **Install Now**.
4. After the plug-in is installed, choose **Settings > Hacklog Remote Attachment**.
5. In the **Hacklog Remote Attachment Options** dialog box, set the FTP service information.

Parameter	Description
FTP server	Specifies the IP address that runs ossftp. We recommend that you set this parameter to 127.0.0.1.
FTP server port number	Specifies the port number of the FTP server. The default value is 2048.
FTP account	Specifies the FTP account in the AccessKeyID/BucketName format. Note that the forward slash (/) is a delimiter and does not indicate an alternative.
FTP password	Specifies the FTP password, which is the AccessKey secret.
FTP timeout	Specifies the FTP timeout period. The default value is 30 seconds.

Parameter	Description
Remote basic URL	Specifies the public endpoint of the bucket. The format is http://BucketName.Endpoint. In this example, the bucket resides in the China (Hangzhou) region, and the bucket name is test-hz-jh-002. Therefore, the URL is http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/wp. For more information about endpoints, see OSS domain names.
FTP remote path	Specifies the path for storing the attachments in the bucket. In the example, wp indicates that all attachments are stored in the wp directory in the bucket. The remote basic URL must correspond to the FTP remote path.
HTTP remote path	Specifies the HTTP remote path. We recommend that you set this parameter to a period (.).

6. Click **Save**.

   A test on the configuration is triggered when you click **Save**. The test result is shown in the upper part of the page.

7. Post a new article to verify whether the configuration is successful.
   i. After you write a new article, click **Add Media** to upload an attachment.
   ii. Click **Post**. You can view the article that you have just written.
   iii. Right-click the image and choose **Open Link in New Tab** from the shortcut menu.

6.4. FAQ

This topic describes the possible problems you may encounter when you use ossftp and their solutions.

- Failed to connect to the FTP server

  Possible causes:

  - The AccessKey ID and AccessKey secret are incorrect.

    Solution: Enter the correct information and try again.

- The AccessKey information is the AccessKey pair of the RAM user account. The RAM user account is not authorized to list buckets.

  Solution: When you use a RAM user account for access, specify the bucket endpoint on the control page for the FTP server. At the same time, the RAM user account also needs some necessary permissions. For more information about how to use RAM to control access to OSS, see Overview. Details:

  - Read-only access

    To obtain the read-only permission when using ossftp, RAM users must obtain permissions to perform ListObjects, GetObject, and HeadObject. For more information about how to configure RAM policy, see Overview.

  - Upload objects

    To upload objects, RAM users must obtain the permission to perform PutObject.

  - Delete objects

    To delete objects, RAM users must obtain the permission to perform DeleteObject.

- 501 and the corresponding error message are returned when you use FileZilla to transfer data after you run the FTP server in Linux

  ```
  501 can't decode path (server filesystem encoding is ANSI_X3.4-1968)
  ```

  The possible cause is that the Chinese character is not properly encoded. Run the following command on the terminal that runs start.sh and restart the terminal:

  ```
  $ export LC_ALL=en_US.UTF-8; export LANG="en_US.UTF-8"; locale
  ```

- Disconnections due to timeout that occurs when you list objects after you log on to ossftp

  The possible cause is that the root directory of the bucket contains an excessive number of files or folders. After you log on to ossftp, the FTP server tries to list all objects and folders in the root directory of the bucket. A maximum of 1,000 objects and folders can be listed each time. If the root directory contains more than 1 million objects and folders, more than 1,000 HTTP requests are sent, which causes timeout.

- Data fails to be transferred due to the port limit of the machine that runs the FTP server

  The control port and data port for FTP are different. When the FTP server needs to transfer data in passive mode, a port is randomly selected and enabled for connections to the client. When the machine where the FTP server resides has port limits, data may fail to be transferred.

  Solution: When ftpserver.py runs, specify the `--passive_ports_start` and `--passive_ports_end` parameters to set the range of the local port and enable the ports that fall within the range.

- Frequent disconnections between the client and FTP server

  Each FTP client has timeout configurations. You can disable timeout configurations or set the timeout value to 0. Take FileZilla as an example. Choose Settings > Connection. Set the timeout value to 0.

7.osscmd (unavailable)

7.1. Overview

osscmd is a Python 2.x-based command line tool. You can use this tool to manage buckets and objects.

> **Note** Commands supported by the osscmd tool have been integrated with the ossutil tool. The osscmd tool is no longer available for downloads as of July 31, 2019.

Scenarios

You can use the osscmd tool in the following scenarios:

- API-based development and debugging. You can use the osscmd tool to send a request in a specific format and perform multipart upload step by step.
- Bucket-based configurations. You can use the osscmd tool to configure logging, website, and lifecycle rules for buckets.

Limits

- The osscmd tool supports Python versions 2.5, 2.6, and 2.7 only.
- The osscmd tool is developed based on Python SDK 0.x. However, Python SDK 2.x.x instead of Python SDK 0.x is maintained.
- Only bugs of the osscmd tool can be fixed. You cannot use the tool to configure new features such as the storage class of infrequent access (IA) or Archive, cross-region replication (CRR), and back-to-origin.

Use the osscmd tool

After you have downloaded and decompressed the Python SDK, run the `python osscmd + operation` command in the directory where the osscmd tool resides. For example, run the following command to upload a file to a bucket:

```
python osscmd put myfile.txt oss://mybucket
```

> **Note** In the commands that are supported by the osscmd tool, oss://bucket specifies a bucket. oss://bucket/object specifies a bucket or an object. oss:// is only a format used to specify resources.

To obtain a detailed list of commands, run the `python osscmd` command.

To obtain a detailed list of command parameters, run the `python osscmd help` command.

7.2. Quick start

This topic describes how to use the osscmd tool.

> **Notice** Commands supported by the osscmd tool have been integrated with the ossutil tool. The osscmd tool is no longer available for download as of July 31, 2019.

Use the osscmd tool

After you have downloaded and decompressed the SDK installation package, you can use the osscmd tool.

You can call Python osscmd to obtain instructions about how to run the commands supported by the osscmd tool. You can use either of the following methods to run each command. The following example shows how to run the gs command to obtain buckets created by a specific user:

- Method 1: The osscmd tool reads the AccessKey ID and AccessKey Secret from the default file. In this case, you do not need to specify the AccessKey ID and AccessKey Secret.

  ```
  $ python osscmd gs
  can't get accessid/accesskey, setup use : config --id=accessid --key=accesskey
  ```

  > **Note** If a similar output is displayed, the AccessKey ID and AcccessKey Secret fail to be read. For more information about how to specify the AccessKey ID and AccessKey Secret, see Method 2.

  Ensure that the AccessKey ID and AccessKey Secret are valid. After you specify the AccessKey ID and AccessKey Secret, run the following command:

  ```
  $ python osscmd gs
  2013-07-19 08:11 test-oss-sample
  Bucket Number is: 1
  ```

- Method 2: Specify the AccessKey ID and AccessKey Secret in the command. The osscmd tool reads the AccessKey ID and AccessKey Secret from the command line. If the AccessKey ID and AccessKey Secret are valid, run the following command:

  ```
  $ python osscmd gs --id=your_id --key=your_key --host=your_endpoint
  2013-07-19 08:11 test-oss-sample
  Bucket Number is: 1
  ```

  To configure the AccessKey ID and AccessKey Secret and import them to the default file, run the following command. The default domain name of OSS is oss.aliyuncs.com.

  ```
  $python osscmd config --id=your_id --key=your_key --host=your_endpoint
  ```

  If a similar output Your configuration is saved into is displayed, the AccessKey ID and AccessKey Secret have been saved.

Basic operations

- List created buckets

  ```
  $python osscmd getallbucket
  ```

  No buckets are displayed if the user has no buckets in OSS.

- Create a bucket

Create a bucket named mybucketname.

```
$python osscmd createbucket mybucketname
```

The bucket may fail to be created. The cause is that the name of a bucket must be unique in OSS. You need to use another bucket name. For example, you can add a specific date to the bucket name.

- Check whether a bucket is created

```
$python osscmd getallbucket
```

If a bucket fails to be created, check the error message returned from the osscmd tool.

- View objects

After a bucket is created, you can view the objects in the bucket.

```
$python osscmd list oss://mybucketname/
```

If there are no objects in the bucket, no objects are displayed.

- Upload a file

Upload a local file to a bucket. If a local file is named local_existed_file, run the following commands to calculate the MD5 value of the file and upload the local file to the bucket:

```
$ md5sum local_existed_file 7625e1adc3a4b129763d580ca0a78e44 local_existed_file
$ python osscmd put local_existed_file oss://mybucketname/test_object
```

> **Note** `md5sum` runs in Linux only.

- View objects again

You can view objects that have been uploaded to a bucket.

```
$python osscmd list oss://mybucketname/
```

- Download an object

Download an object from a bucket to a local file. Compare the MD5 value of the object that is downloaded with the object MD5 value calculated before the object is downloaded.

```
$ python osscmd get oss://mybucketname/test_object download_file
$ md5sum download_file
7625e1adc3a4b129763d580ca0a78e44  download_file
```

> **Note** `md5sum` runs in Linux only.

- Delete an object

$ python osscmd delete oss://mybucketname/test_object

- Delete a bucket

> **Note** If a bucket contains objects, the bucket cannot be deleted.

```
$ python osscmd deletebucket mybucketname
```

Use lifecycle

- Create a TXT file and save it as an XML file. Use the XML file to configure lifecycle rules.

```
<LifecycleConfiguration>
    <Rule>
        <ID>1125</ID>
        <Prefix>log_backup/</Prefix>
        <Status>Enabled</Status>
        <Expiration>
            <Days>2</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

If you run the preceding code, objects whose names are prefixed with log_backup/ and that are retained for two days from their last modification time are deleted. For more information about the configuration of lifecycle rules, see API Reference.

- Write a lifecycle rule.

```
python osscmd putlifecycle oss://mybucket lifecycle.xml
0.150(s) elapsed
```

- Read a lifecycle rule.

```
python osscmd getlifecycle oss://mybucket
<? xml version="1.0" encoding="UTF-8"? >
<LifecycleConfiguration>
  <Rule>
    <ID>1125</ID>
    <Prefix>log_backup/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>2</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
0.027(s) elapsed
```

- Delete a lifecycle rule.

```
python osscmd deletelifecycle oss://mybucket
0.139(s) elapsed
```

- Read a lifecyle rule.

```
python osscmd getlifecycle oss://mybucket
Error Headers:
[('content-length', '288'), ('server', 'AliyunOSS'), ('connection', 'close'), ('x-oss-req
uest-id', '54C74FEE5D7F6B24E5042630'), ('date', 'Tue, 27 Jan 2015 08:44:30 GMT'), ('conte
nt-type', 'application/xml')]
Error Body:
<? xml version="1.0" encoding="UTF-8"? >
<Error>
  <BucketName>mybucket</BucketName>
  <Code>NoSuchLifecycle</Code>
  <Message>No Row found in Lifecycle Table. </Message>
  <RequestId>54C74FEE5D7F6B24E5042630</RequestId>
  <HostId>mybucket.oss-maque-hz-a.alibaba.net</HostId>
</Error>
Error Status:
404
getlifecycle Failed!
```

Configure hotlinking protection

- Allow access from a request that has an empty Referer field.

```
$osscmd putreferer oss://test --allow_empty_referer=true
0.004(s) elapsed
```

- Obtain the Referer whitelist.

```
$osscmd getreferer oss://test
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
  <AllowEmptyReferer>true</AllowEmptyReferer>
  <RefererList />
</RefererConfiguration>
```

- The Referer field is required. Only requests that have the Referer field value of www.example.com are allowed.

```
$osscmd putreferer oss://test --allow_empty_referer=false --referer='www.example.com'
0.092(s) elapsed
```

- Obtain the Referer whitelist.

```
$osscmd getreferer oss://test
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
  <AllowEmptyReferer>false</AllowEmptyReferer>
  <RefererList>
    <Referer>www.example.com</Referer>
  </RefererList>
</RefererConfiguration>
```

- The Referer field is required. Only requests that have the Referer field value of www.example.com or www.example.org are allowed.

```
$osscmd putreferer oss://test --allow_empty_referer=false --referer='www.example.com,www.
example.org'
```

- Obtain the Referer whitelist.

```
$osscmd getreferer oss://test
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
  <AllowEmptyReferer>false</AllowEmptyReferer>
  <RefererList>
    <Referer>www.example.com</Referer>
    <Referer>www.example.org</Referer>
  </RefererList>
</RefererConfiguration>
```

Use logging

- Configure a logging rule

```
$osscmd putlogging oss://mybucket oss://myloggingbucket/mb
```

- Obtain logging rules that are configured for a bucket

```
$osscmd getlogging oss://mybucket
```

7.3. Commands for operations on buckets

This topic describes commands that can be used to manage buckets.

> **Notice** Commands supported by the osscmd tool have been integrated with the ossutil tool. The osscmd tool is no longer available for downloads as of July 31, 2019.

config

Command:

```
config --id=[accessid] --key=[accesskey] --host=[host] --sts_token=[sts_token]
```

Example:

- `python osscmd config --id=your_id --key=your_key`

```
python osscmd config --id=your_id --key=your_key
       --host=oss-internal.aliyuncs.com
```

getallbucket(gs)

Command:

```
getallbucket(gs)
```

Obtain created buckets. gs is short for get allbucket. You can run the gs or allbucket command to obtain a list of created buckets.

Example:

- `python osscmd getallbucket`
- `python osscmd gs`

createbucket(cb,mb,pb)

Command:

```
createbucket(cb,mb,pb) oss://bucket --acl=[acl]
```

Create a bucket.

- cb is short for create bucket. mb is short for make bucket. pb is short for put bucket.
- You can set oss://bucket to specify a bucket name.
- The acl parameter is optional.

Example:

- `python osscmd createbucket oss://mybucket`
- `python osscmd cb oss://myfirstbucket --acl=public-read`
- `python osscmd mb oss://mysecondbucket --acl=private`
- `python osscmd pb oss://mythirdbucket`

deletebucket(db)

Command:

```
deletebucket(db) oss://bucket
```

Delete a bucket. db is short for delete bucket.

Example:

- `python osscmd deletebucket oss://mybucket`
- `python osscmd db oss://myfirstbucket`

deletewholebucket

> **Warning** All data is deleted if you run this command. Deleted data cannot be recovered. Exercise caution when you run this command.

Command:

```
deletewholebucket oss://bucket
```

Delete a bucket, and all objects and fragments in the bucket.

Example:

```
python osscmd deletewholebucket oss://mybucket
```

getacl

Command:

```
getacl oss://bucket
```

Obtain the bucket ACL.

Example:

```
python osscmd getacl oss://mybucket
```

setacl

Command:

```
setacl oss://bucket --acl=[acl]
```

Modify the bucket ACL. You can set the bucket ACL to private, public-read, or public-read-write.

Example:

```
python osscmd setacl oss://mybucket --acl=private
```

putlifecycle

Command:

```
putlifecycle oss://mybucket lifecycle.xml
```

Set lifecycle rules. In the command, lifecycle.xml indicates a file that is used to configure lifecycle rules. For more information, see API Reference.

Example:

```
python osscmd putlifecycle oss://mybucket lifecycle.xml
```

Example:

```
<LifecycleConfiguration>
    <Rule>
        <ID>1125</ID>
        <Prefix>log_backup/</Prefix>
        <Status>Enabled</Status>
        <Expiration>
            <Days>2</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

getlifecycle

Command:

```
osscmd getlifecycle oss://bucket
```

Obtain lifecycle rules of a bucket.

Example:

```
python osscmd getlifecycle oss://mybucket
```

deletelifecycle

Command:

```
osscmd deletelifecycle oss://bucket
```

Delete all lifecycle rules of a bucket.

Example:

```
python osscmd deletelifecycle oss://mybucket
```

putreferer

Command:

```
osscmd putreferer oss://bucket --allow_empty_referer=[true|false]
        --referer=[referer]
```

Set hotlinking protection rules. The `allow_empty_referer` parameter is required and is used to specify whether an empty Referer field is allowed. The `referer` parameter is used to set the Referer whitelist. For example, you can add www.example.com and www.example.org to the Referer whitelist. To add multiple domain names, separate the domain names with commas (,). For more information about configuration rules, see Configure hotlinking protection.

Example:

```
python osscmd putreferer oss://mybucket --allow_empty_referer=true
          --referer="www.example.com,www.example.org"
```

getreferer

Command:

```
osscmd getreferer oss://bucket
```

Obtain the hotlinking protection rule of the bucket.

Example:

```
python osscmd getreferer oss://mybucket
```

putlogging

Command:

```
osscmd putlogging oss://source_bucket oss://target_bucket/[prefix]
```

source_bucket specifies the bucket that is accessed. target_bucket specifies the bucket that is used to store the log of access to the source bucket. You can set a prefix for the log that is generated to record access to the source bucket and facilitate log queries.

Example:

```
python osscmd getlogging oss://mybucket
```

getlogging

Command:

```
osscmd getlogging oss://bucket
```

Obtain the access log setting rule of the bucket.

Example:

```
python osscmd getlogging oss://mybucket
```

7.4. Commands for operations on objects

This topic describes commands that can be used to manage objects.

> **Notice** Commands supported by the osscmd tool have been integrated with the ossutil tool. The osscmd tool is no longer available for download as of July 31, 2019.

ls(list)

Command:

```
ls(list) oss://bucket/[prefix] [marker] [delimiter] [maxkeys]
```

List objects in a bucket. You can specify a prefix to list all objects whose names start with the specified prefix. For example, you can specify abc as the prefix to list all objects whose names start with abc.

Example:

- `python osscmd ls oss://mybucket/folder1/folder2`
- `python osscmd ls oss://mybucket/folder1/folder2 marker1`
- `python osscmd ls oss://mybucket/folder1/folder2 marker1 /`
- `python osscmd ls oss://mybucket/`
- `python osscmd list oss://mybucket/ "" "" 100`

Command:

```
ls(list) oss://bucket/[prefix] --marker=xxx --delimiter=xxx --maxkeys=xxx
        --encoding_type=url
```

List objects in a bucket. You can set encoding_type to specify the encoding method that is used during transmission. If you set encoding_type to url, objects whose names contain control characters are encoded.

Example:

- `python osscmd ls oss://mybucket/folder1/folder2 --delimiter=/`
- `python osscmd ls oss://mybucket/folder1/folder2 --marker=a`
- `python osscmd ls oss://mybucket/folder1/folder2 --maxkeys=10`

mkdir

Command:

```
mkdir oss://bucket/dirname
```

Create a folder.

Example:

```
python osscmd mkdir oss://mybucket/folder
```

listallobject

Command:

```
listallobject oss://bucket/[prefix]
```

List all objects in a bucket. You can specify a prefix to list objects whose names start with the prefix.

Example:

- `python osscmd listallobject oss://mybucket`
- `python osscmd listallobject oss://mybucket/testfolder/`

deleteallobject

Command:

`deleteallobject oss://bucket/[prefix]`

Delete all objects in a bucket. You can also specify a prefix to delete objects whose names start with the prefix.

Example:

- `python osscmd deleteallobject oss://mybucket`
- `python osscmd deleteallobject oss://mybucket/testfolder/`

downloadallobject

Command:

```
downloadallobject oss://bucket/[prefix] localdir --replace=false
      --thread_num=5
```

Download objects from a bucket to a local directory. This operation ensures that the original directory structure remains the same. You can specify a prefix to download objects whose names start with the specified prefix. `--replace=false` indicates that local files with the same name of the object will not be overwritten during the download. `--replace=true` indicates that local files with the same name of the object will be overwritten. You can also use thread_num to configure the download thread.

Example:

- `python osscmd downloadallobject oss://mybucket /tmp/folder`

  ```
  python osscmd downloadallobject oss://mybucket /tmp/folder
         --replace=false
  ```

  ```
  python osscmd downloadallobject oss://mybucket /tmp/folder --replace=true
           --thread_num=5
  ```

downloadtodir

Command:

`downloadtodir oss://bucket/[prefix] localdir --replace=false`

Download objects from a bucket to a local directory. This operation ensures that the original directory structure remains the same. You can specify a prefix to download objects whose names start with the specified prefix. `--replace=false` indicates that local files with the same name of the object will not be overwritten during the download. `--replace=true` indicates that local files with the same name of the object will be overwritten. downloadtodir follows the same logic as that of downloadallobject.

Example:

- `python osscmd downloadtodir oss://mybucket /tmp/folder`

- `python osscmd downloadtodir oss://mybucket /tmp/folder --replace=false`

```
python osscmd downloadtodir oss://mybucket /tmp/folder
      --replace=true
```

uploadfromdir

Command:

```
uploadfromdir localdir oss://bucket/[prefix] --check_point=check_point_file --replace=false
        --check_md5=false --thread_num=5
```

Upload local files to a bucket.

If local directory `/tmp/` contains the a/b, a/c, and a files, the paths of these files in OSS are oss://bucket/a/b, oss://bucket/a/c, and oss://bucket/a. If a prefix is set to mytest, the paths of these files in OSS are oss://bucket/mytest/a/b, oss://bucket/mytest/a/c, and oss://bucket/mytest/a.

`--check_point=check_point_file` is used to specify a checkpoint file. After the checkpoint file is specified, the osscmd tool will be used to store the timestamps that are recorded when the local files are uploaded. The uploadfromdir command is used to compare the timestamps of the files that are being uploaded and the timestamps that are recorded in the checkpoint file. If the timestamps are different, the files are reuploaded. check_point_file is not specified by default. --replace=false indicates that local files with the same name of the object will not be overwritten during the upload. `--replace=true` indicates that local files with the same name of the object will be overwritten. `--check_md5=false` indicates that Content-MD5 is not included in the request header and MD5 verification will not be performed. --check_md5=true indicates that MD5 verification will be performed.

Note: The checkpoint file stores upload records of all objects.

Example:

- `python osscmd uploadfromdir /mytemp/folder oss://mybucket`

```
python osscmd uploadfromdir /mytemp/folder oss://mybucket
          --check_point_file=/tmp/mytemp_record.txt
```

```
python osscmd uploadfromdir C:\Documents and Settings\User\My Documents\Downloads
          oss://mybucket --check_point_file=C:\cp.txt
```

put

Command:

```
put localfile oss://bucket/object --content-type=[content_type]
        --headers="key1:value1#key2:value2" --check_md5=false
```

When uploading a local file to a bucket, you can set HTTP header fields such as content-type. `--check_md5=false` indicates that Content-MD5 is not included in the request header and MD5 verification will not be performed. --check_md5=true indicates that MD5 verification will be performed.

Example:

- `python osscmd put myfile.txt oss://mybucket`

- `python osscmd put myfile.txt oss://mybucket/myobject.txt`

```
python osscmd put myfile.txt oss://mybucket/test.txt --content-type=plain/text
          --headers="x-oss-meta-des:test#x-oss-meta-location:CN"
```

```
python osscmd put myfile.txt oss://mybucket/test.txt
        --content-type=plain/text
```

upload

Command:

```
upload localfile oss://bucket/object --content-type=[content_type]
      --check_md5=false
```

Upload local files to a bucket. `--check_md5=false` indicates that Content-MD5 is not included in the request header and MD5 verification will not be performed. --check_md5=true indicates that MD5 verification will be performed.

Example:

```
python osscmd upload myfile.txt oss://mybucket/test.txt
        --content-type=plain/text
```

get

Command:

```
get oss://bucket/object localfile
```

Download an object to a local file.

Example:

```
python osscmd get oss://mybucket/myobject /tmp/localfile
```

multiget(multi_get)

Command:

```
multiget(multi_get) oss://bucket/object localfile --thread_num=5
```

Use multithreading to download an object to a local file. You can configure the number of threads that are used to download the object.

Example:

- `python osscmd multiget oss://mybucket/myobject /tmp/localfile`
- `python osscmd multi_get oss://mybucket/myobject /tmp/localfile`

cat

Command:

```
cat oss://bucket/object
```

Read and display object content. Do not run this command if the object is large.

Example:

```
python osscmd cat oss://mybucket/myobject
```

meta

Command:

```
meta oss://bucket/object
```

Read and display the meta information of the object. Meta information contains the content-type, file length, and user metadata.

Example:

```
python osscmd meta oss://mybucket/myobject
```

copy

Command:

```
copy oss://source_bucket/source_object oss://target_bucket/target_object
        --headers="key1:value1#key2:value2"
```

Replicate an object from a source bucket to a destination bucket.

Example:

```
python osscmd copy oss://bucket1/object1 oss://bucket2/object2
```

rm(delete,del)

Command:

```
rm(delete,del) oss://bucket/object --encoding_type=url
```

Delete an object. When encoding-type is set to url, control characters to be deleted also need to be URL-encoded.

Example:

- `python osscmd rm oss://mybucket/myobject`
- `python osscmd delete oss://mybucket/myobject`
- `python osscmd del oss://mybucket/myobject`
- `python osscmd del oss://mybucket/my%01object --encoding_type=url`

signurl(sign)

Command:

```
signurl(sign) oss://bucket/object --timeout=[timeout_seconds]
```

Generate a signed URL containing the timeout value. A signed URL is used to provide access to a specific object when the bucket ACL is private.

Example:

- `python osscmd sign oss://mybucket/myobject`
- `python osscmd signurl oss://mybucket/myobject`

7.5. Commands for operations on parts

This topic describes commands that can be used to manage parts.

> **Notice** Commands supported by the osscmd tool have been integrated with the ossutil tool. The osscmd tool is no longer available for downloads as of July 31, 2019.

init

Command:

```
init oss://bucket/object
```

Initialize an upload event to generate an upload ID. You can add this upload ID to the multiupload command to perform operations on parts.

Example:

```
python osscmd init oss://mybucket/myobject
```

listpart

Command:

```
listpart oss://bucket/object --upload_id=xxx
```

List the parts that are uploaded by using the upload ID of a specified object. For more information about related concepts, see OSS API Reference. You must specify the upload ID.

Example:

```
python osscmd listpart oss://mybucket/myobject --upload_id=
          75835E389EA648C0B93571B6A46023F3
```

listparts

Command:

```
listparts oss://bucket
```

List the objects and upload IDs of multipart upload events that have not been completed for a bucket. When you want to delete a bucket but the system prompts that the bucket is not empty, you can run this command to check whether there are fragments in the bucket.

Example:

```
python osscmd listparts oss://mybucket
```

getallpartsize

Command:

```
getallpartsize oss://bucket
```

List the total size of parts that are uploaded by using the existing upload IDs.

Example:

```
python osscmd getallpartsize oss://mybucket
```

cancel

Command:

```
cancel oss://bucket/object --upload_id=xxx
```

Terminate the multipart upload event that uses the upload ID.

Example:

```
python osscmd cancel oss://mybucket/myobject --upload_id=
          D9D278DB6F8845E9AFE797DD235DC576
```

multiupload(multi_upload,mp)

Command:

```
multiupload(multi_upload,mp) localfile oss://bucket/object --check_md5=false
        --thread_num=10
```

Use multipart upload to upload a local file to OSS.

Example:

- `python osscmd multiupload /tmp/localfile.txt oss://mybucket/object`
- `python osscmd multiup_load /tmp/localfile.txt oss://mybucket/object`
- `python osscmd mp /tmp/localfile.txt oss://mybucket/object`

Command:

```
multiupload(multi_upload,mp) localfile oss://bucket/object --upload_id=xxx --thread_num=10
        --max_part_num=1000 --check_md5=false
```

Use multipart upload to upload a local file to OSS. The part count of the local file is defined by the max_part_num parameter. When this command is run, the system first determines whether the MD5 value of ETags of parts that use the upload ID is the same with the MD5 value of the local file. If their values are the same, the parts are uploaded. Generate an upload ID before this upload event is started. Add the upload ID to the command. If the upload fails, you can run the same multiupload command to upload the parts in the same way you use resumable upload. `--check_md5=false` indicates that Content-MD5 is not included in the request header and MD5 verification will not be performed. --check_md5=true indicates that MD5 verification will be performed.

Example:

```
python osscmd multiupload /tmp/localfile.txt oss://mybucket/object --upload_id=
          D9D278DB6F8845E9AFE797DD235DC576
```

```
python osscmd multiup_load /tmp/localfile.txt oss://mybucket/object
        --thread_num=5
```

- `python osscmd mp /tmp/localfile.txt oss://mybucket/object --max_part_num=100`

copylargefile

Command:

```
copylargefile oss://source_bucket/source_object oss://target_bucket/target_object
        --part_size=10*1024*1024 --upload_id=xxx
```

To replicate an object that is larger than 1 GB, use multipart to replicate the object to the destination bucket. Ensure that the source bucket and destination bucket are in the same region. The upload_id parameter is optional. If you need to resume the transmission of a multipart copy event, you can import the upload_id parameter for the multipart copy event. The part_size parameter is used to define the size of each part. A single part must be at least 100 KB in size. A maximum of 10,000 parts are supported for a multipart copy event. If the value of part_size is smaller than 100 KB, the program automatically adjusts the part size.

Example:

```
python osscmd copylargefile oss://source_bucket/source_object
          oss://target_bucket/target_object --part_size=10*1024*1024
```

uploadpartfromfile (upff)

Command:

```
uploadpartfromfile (upff) localfile oss://bucket/object --upload_id=xxx
        --part_number=xxx
```

This command is used for tests only.

uploadpartfromstring(upfs)

Command:

```
uploadpartfromstring(upfs) oss://bucket/object --upload_id=xxx --part_number=xxx
        --data=xxx
```

This command is used for tests only.

8.RAM Policy Editor

This topic describes how to use RAM Policy Editor.

Address

RAM Policy Editor

Usage

A RAM policy consists of several rules. By using RAM Policy Editor, you can add or delete rules for a policy and generate the JSON text for the policy in a graphical interface. You can copy the JSON text generated for a policy in RAM Policy Editor to the Policy Document section in the RAM console to create the RAM policy.

For more information, see Create a custom policy.

To add a rule to a policy in RAM Policy Editor, you must configure the following fields for the rule: Effect, Actions, Resources, and Conditions.

- Effect

  Specifies whether the rule allows or denies access to resources.

- Actions

  Specifies the actions performed to access resources. In most cases, you only need to select from the following general actions provided in the drop-down list:

  - `oss:*` : indicates all actions are allowed or denied.
  - `oss:Get*` : indicates all read actions are allowed or denied.
  - `oss:Put*` : indicates all write actions are allowed or denied.

  For more information, visit RAM Policy Editor README.

- Resources

  Specifies the OSS resources that the rule allows or denies to access. You can specify multiple resources in the following formats:

  - `my-bucket` : indicates a bucket. If you specify the resource in this format, users have permissions to perform actions on the bucket but not the objects in the bucket.
  - `my-bucket/*` : indicates all objects in a bucket. If you specify the resource in this format, users have permissions to perform actions on the objects in the bucket but not the bucket. For example, the ListObjects operation cannot be performed on the bucket.
  - `my-bucket/dir` : indicates a directory in a bucket. If you specify the resource in this format, users have permissions to perform actions on the directory but not the objects in the directory.
  - `my-bucket/dir/*` : indicates all objects in a directory. If you specify the resource in this format, users have permissions to perform actions on the objects in the directory but not the directory. For example, the ListObjects operation cannot be performed on the directory.
  - `acs:oss:*:1234:my-bucket/dir` : the complete resource path. In this example, `1234` indicates the user ID that can be viewed in the console.

  EnablePath

  To grant users permissions to access a directory, you need also to grant them permissions to perform

ListObjects on the parent directory. For example, to ensure that users can view a directory named my-bucket/users/dir/ in the OSS console or other tools, in addition to specifying resources in the `my-bucket/users/dir/*` format, you must also grant the users the following permissions:

```
ListObjects my-bucket
ListObjects my-bucket/users
ListObjects my-bucket/users/dir
```

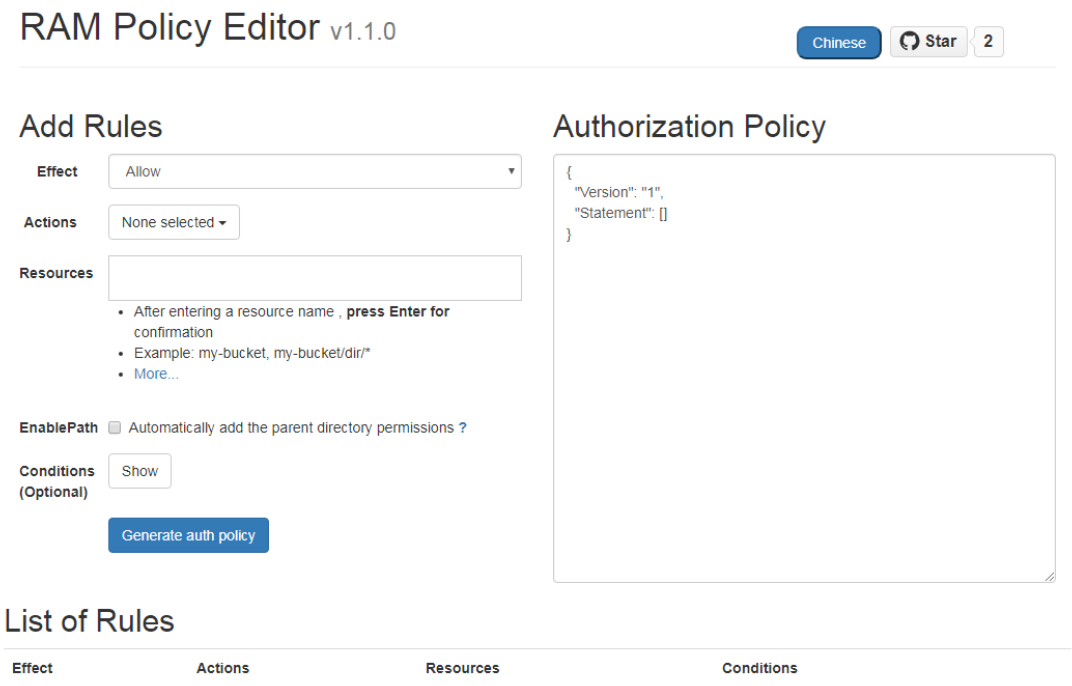
If you select EnablePath, the preceding permissions are automatically granted.

- Conditions

  Specifies the conditions that must be met to access the resources. You can specify multiple conditions in a policy.

Example

The following figure shows a policy that you can configure to grant users complete permissions on the `my-bucket` bucket and all objects in the bucket.



For more examples on how to use RAM Policy Editor, visit RAM Policy Editor README.