

Alibaba Cloud Object Storage Service

Tools

Issue: 20190920

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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, 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 the Alibaba Cloud website, product programs, or content 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 contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information 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.
	This warning information 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 restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	It is used for commands.	Run the <code>cd / d C :/ windows</code> command to enter the Windows system folder.
<i>Italics</i>	It is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	It indicates that it is an optional value, and only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
<code>{}</code> or <code>{a b}</code>	It indicates that it is a required value, and only one item can be selected.	<code>swich {stand slave}</code>

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 OSS tools.....	1
2 ossbrowser.....	5
2.1 Quick start.....	5
2.2 Permission management.....	13
3 ossutil.....	19
3.1 Overview.....	19
3.2 Download and installation.....	21
3.3 Common commands.....	23
3.3.1 appendfromfile.....	23
3.3.2 bucket-encryption.....	25
3.3.3 bucket-policy.....	27
3.3.4 bucket-tagging.....	30
3.3.5 cat.....	32
3.3.6 config.....	33
3.3.7 cors.....	37
3.3.8 cors-options.....	39
3.3.9 cp.....	41
3.3.10 create-symlink.....	57
3.3.11 du.....	58
3.3.12 getallpartsize.....	60
3.3.13 hash.....	61
3.3.14 help.....	63
3.3.15 lifecycle.....	63
3.3.16 listpart.....	66
3.3.17 logging.....	67
3.3.18 ls.....	69
3.3.19 mb.....	74
3.3.20 mkdir.....	76
3.3.21 object-tagging.....	77
3.3.22 probe.....	81
3.3.23 read-symlink.....	85
3.3.24 referer.....	86
3.3.25 restore.....	89
3.3.26 request-payment.....	91
3.3.27 rm.....	93
3.3.28 set-acl.....	96
3.3.29 set-meta.....	99
3.3.30 sign.....	102

3.3.31 stat.....	104
3.3.32 update.....	105
3.3.33 website.....	107
3.4 View all supported options.....	111
3.5 Bucket-related commands.....	118
3.6 Object-related commands.....	133
3.7 Multipart-related commands.....	139
4 ossimport.....	143
4.1 Architecture and configuration.....	143
4.2 Standalone deployment.....	162
4.3 Distributed deployment.....	167
4.4 FAQ.....	172
5 RAM Policy Editor.....	181
6 ossftp.....	184
6.1 Quick installation for OSS FTP.....	184
6.2 How to store remote attachments to your OSS instance with Discuz.....	189
6.3 How to store remote attachments to your OSS instance with PHPWind....	191
6.4 How to store remote attachments to your OSS instance with WordPress...	192
6.5 How to integrate RAM for file sharing.....	194
6.6 FAQ.....	198
7 ossfs.....	200
7.1 Quick installation.....	200
7.2 FAQ.....	202
8 osscmd (unavailable).....	210
8.1 Overview.....	210
8.2 Quick start.....	211
8.3 Commands for operations on buckets.....	216
8.4 Commands for operations on objects.....	221
8.5 Commands for operations on parts.....	228

1 OSS tools

In addition to the OSS console, you can use the following tools to use OSS more efficiently.

Tool	Description
ossbrowser	<p>A graphical object management tool.</p> <ul style="list-style-type: none">• 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 folders.• Allows you to use concurrent upload and resumable upload to upload files.• Allows you to configure authorization policies and grant RAM users permissions.• Supports Windows, Linux, and Mac. <p>Limits:</p> <ul style="list-style-type: none">• The transmission speed and performance of ossbrowser are not as good as those of ossutil because ossbrowser is a graphical tool.• Only files smaller than 5 GB can be uploaded or replicated.
ossutil	<p>A command line tool that is used to manage objects and buckets.</p> <ul style="list-style-type: none">• Provides a wide range of convenient and simple commands to manage objects and buckets while ensuring good operation performance.• Allows you to use concurrent upload and resumable upload to upload files.• Allows you to upload and download folders.• Supports typical bucket management-related commands.

Tool	Description
osscmd (unavailable)	<p>A command line tool that is used to manage objects and buckets.</p> <ul style="list-style-type: none">• Provides a wide range of commands to manage objects and buckets.• Supports Windows and Linux. <p>Limits:</p> <ul style="list-style-type: none">• The osscmd tool is compatible with Python versions 2.5, 2.6, and 2.7 only.• 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. <div data-bbox="534 869 1439 1070"> 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.</div>

Tool	Description
ossfs	<p>A tool that is used to attach a bucket to the local file system.</p> <p>After you attach OSS buckets to the local file system of Linux, you can perform operations on the objects in OSS through the local file system to access or share these objects.</p> <ul style="list-style-type: none"> • Supports most functions of POSIX-based file systems, including file read/write, directories, soft links, permissions, UID or GID, and extended attributes. • Allows you to use multipart upload to upload large files. • Supports MD5 verification to ensure data integrity. <p>Limits:</p> <ul style="list-style-type: none"> • You cannot attach an Archive bucket. • If you use ossfs tool to edit an uploaded file, the file is uploaded again. • The performance of metadata-related operations is compromised when you run the <code>list directory</code> command. In this case, you must remotely access the OSS server. • Errors may occur when you rename an object or a folder. Operation failures may cause data inconsistency. • The ossfs tool is not suitable for scenarios where read and write operations are highly concurrent. • You must maintain data consistency when an OSS bucket is attached to multiple clients. For example, you must schedule object use to prevent it from being written by multiple clients at the same time. • Hard links are not supported.
ossftp	<p>An FTP-based tool that is used to manage objects in OSS.</p> <ul style="list-style-type: none"> • You can use FTP clients such as FileZilla, WinSCP, and FlashFXP to manage data in OSS. • The ossftp tool is an FTP server that receives FTP requests and performs operations on objects and folders in OSS. • The ossftp tool is based on Python V2.7 and later. • This tool supports Windows, Linux, and Mac.

Tool	Description
ossimport	<p>A tool that is used to synchronize data to OSS.</p> <ul style="list-style-type: none"> • Allows you to use the ossimport tool to synchronize data from a third-party data source to OSS. • Supports the distributed deployment mode. You can use multiple servers to migrate multiple data simultaneously. • Supports the migration of 1 TB of data at least. • Supports Windows and Linux. • Applies to Java V1.7 and later.
Visual signature tool	<p>A third-party visual signature tool.</p> <ul style="list-style-type: none"> • This tool is used to generate a signed URL for operations on data in OSS. • You can debug the system to resolve problems you encounter when the signed URL is generated. If you encounter any problems when the signed URL is generated, compare the signed URL with the signature of the tool to troubleshoot the problems. • The browser edition of this tool supports the following browsers: Google Chrome, Firefox, and Safari.
RAM Policy Editor	<p>A tool that is used to achieve automatic generation of OSS-related authorization policies.</p> <ul style="list-style-type: none"> • This tool can automatically generate authorization policies based on specified information. You can add the generated policies to the custom policy in the RAM console. • This tool supports the following browsers: Google Chrome, Firefox, and Safari. <div data-bbox="539 1464 1434 1626" style="background-color: #f0f0f0; padding: 5px;">  Note: We recommend that you use this tool to generate custom authorization policies. </div>
oss-emulator	<p>A lightweight simulator of OSS.</p> <ul style="list-style-type: none"> • The oss-emulator tool provides the same API as that of OSS. This tool can be used to the debug and test OSS functions. • This tool is based on Ruby V2.2.8 and later. • This tool supports Windows and Linux.

2 ossbrowser

2.1 Quick start

Ossbrowser is a graphical management tool developed by Alibaba Cloud. It provides features similar to those of Windows Explorer. Using ossbrowser, you can view, upload, download, and manage objects with ease.



Note:

- You can only move or copy objects smaller than 5 GB by using ossbrowser. For objects larger than 5 GB, we recommend you use [ossutil](#).
- Ossbrowser supports Linux, Mac, and Windows (Windows 7 and later). We recommend you do not use ossbrowser in Windows XP and Windows Server.

Installation

1. Download and install ossbrowser

Supported operating system	Download URL
Windows x32	Windows x32
Windows x64	Windows x64
MAC	MAC
Linux x64	Linux x64



Note:

For more download URLs, see [GitHub](#).

2. Start ossbrowser.

The screenshot shows the 'Token Login' tab of the ossbrowser interface. It contains the following fields and controls:

- Endpoint:** A dropdown menu with 'Default (Public Cloud)' selected.
- AccessKeyId:** A text input field containing a masked key ID.
- AccessKeySecret:** A text input field with masked characters.
- Preset OSS Path:** A text input field containing 'oss://bucketname'.
- request payer:** A checked checkbox.
- Region:** A dropdown menu with 'South China 1(Shenzhen)' selected.
- Description:** A text input field with the placeholder 'Optional, Up to 30 words'.
- Remember:** An unchecked checkbox.
- Login:** A blue button.
- AK Histories:** A grey button.

Set the following parameters to log on to ossbrowser:

- **Endpoint:** Select the region (endpoint) that you want to log on.
 - **Default:** Log on to ossbrowser with the default endpoint.
 - **Customize:** Enter the endpoint you want to use to log on to ossbrowser. You can enter a URL starting with "http" or "https" to log on to ossbrowser through the HTTP or HTTPS method, for example, https://oss-cn-

beijing.aliyuncs.com. For more information about the regions and endpoints, see [Regions and endpoints](#).

- **cname:** You can log on to ossbrowser with a custom domain name (CNAME) attached to your OSS resources. For more information about attaching a CNAME, see [Attach a custom domain name](#).
- **AccessKeyId/AccessKeySecret:** Enter the Accesskey (AK) of your account. To ensure data security, we recommend that you use the AK of a RAM user to log on to ossbrowser. For more information about AK, see [Create an AccessKey](#).
- **Preset OSS Path:**
 - **Administrator RAM users with administration permissions on all buckets:** No configuration is required.
 - **Operator RAM users:** Configurations are required. Enter the path of the OSS bucket or sub-directory that you want to access (the RAM user must have permission to access the OSS bucket or sub-directory). The path format is as follows: `oss:// bucket name/sub-directory name/`.
- **Region:** Select the region where the OSS resources belong to.
- **Remember:** Select to save the AK. When you log on to ossbrowser later, you can simply click AK Histories and select the saved AK instead of entering the AK repeatedly. Do not select this option if you use a shared computer.

Usage

Ossbrowser supports simple management operations on OSS resources.

- Manage a bucket

- Create a bucket.

1. On the main interface of ossbrowser, click Create Bucket.

2. Set the following information about the bucket:

- **Name:** The name of a bucket can be 63 characters in maximum and must be unique.

- **Region:** Select the region where the bucket belongs to.

- **ACL:** Select the ACL for the bucket. For more information about ACL, see [ACL](#).

- **Type:** Select the default storage class of the bucket. For more information about storage class, see [Introduction to storage classes](#).

3. Click OK.

- Delete a bucket.

Select the bucket that you want to delete, and then click More > Remove. A bucket cannot be deleted when objects or parts are stored in it.

- Manage objects/directories

- Create a directory.

1. On the main interface of ossbrowser, click the bucket in which you want to create a folder.

2. Click Directory.

3. Enter the name of the directory and click OK.



Note:

- Emoticons are not allowed in a directory name. Use compliant UTF-8 characters in directory names.

- You can create only a single-level directory at a time. For example, you can create a single-level directory `abc` but not a multi-level directory `abc / 123`.

- A sub-directory named `..` is not allowed.

■ The length of a directory name must be in a range of 1 to 254 characters.

- Upload files/directories.

In the specified bucket or directory, click Files/Folder, and then select the files or folders that you want to upload. You can upload multiple files or folders at the same time.

- Download objects/directories

In the specified bucket or directory, select the objects or directories that you want to download, and then click Download. You can download multiple objects or folders at the same time.

- Copy objects/directories.

1. In the specified bucket or directory, select the objects or directories that you want to copy, and then click Copy.

2. Enter the bucket or directory where you want to copy the data to, and then click Paste. If the source address and target address of the copied object are the same, the original object is overwritten. If the storage class of the overwritten object is IA or Archive and the storage period of the object does not reach the required value, fees incur for the advanced deletion. For more information, see [Billing items](#).

- Move objects/directories.

1. In the specified bucket or directory, select the objects or directories that you want to move, and then click More > Move.

2. Enter the bucket or directory where you want to move the data to, click Paste.



Notice:

When you move an object or a directory, the object or directory is copied from the source address to the target address, and the object or directory in the source address is deleted. If you move an object of the IA or Archive

storage class and the storage period of the object does not reach the required value, fees incur for the advanced deletion.

- Rename objects/directories

In the specified bucket or directory, select the objects or directories that you want to rename, click More > Rename, and then enter the new name.



Notice:

- You can only rename objects smaller than 1 GB.
- When you rename an object or a directory, the object or directory is copied, renamed, and then saved. The original object or directory is deleted. If you rename an object of the IA or Archive storage class and the storage period of the object does not reach the required value, fees incur for the advanced deletion.

- Delete objects/directories

Select the object or directory that you want to delete, and then click More > Remove. If you delete an object of the IA or Archive storage class and the storage period of the object does not reach the required value, fees incur for the advanced deletion.

- Generate an access URL for an object.

1. Select the specified object, and then click More > Address.
2. Enter the valid period of the URL, and then click Generate.
3. Click Copy or Mail it to send the URL to users who want to access the object.
You can also scan the QR code to access the object.

- Preview an object.

You can double-click an object to preview it. You can preview images and objects in the txt and pdf formats in ossbrowser.

- Manage parts.

Select the specified bucket, and then click Multipart. You can delete unnecessary parts.

More operations

- Upload/Download performance optimization

You can click Settings to configure the following parameters.

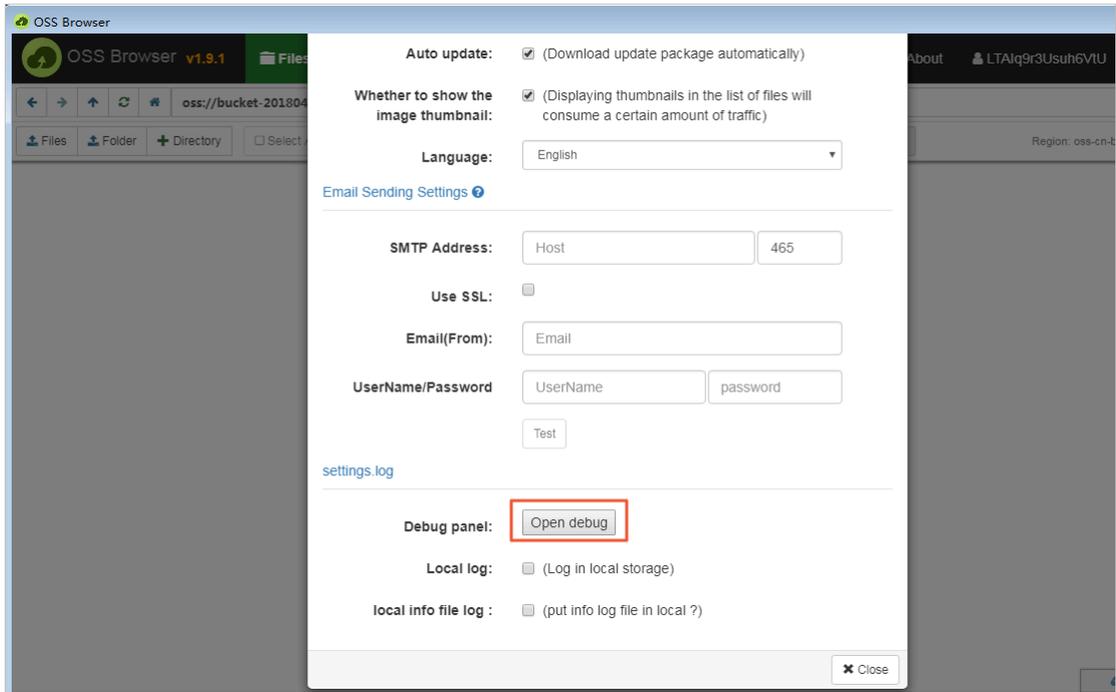
- Upload tasks concurrent number: Specify the maximum number of upload tasks that can be performed at the same time. If the number of upload tasks is larger than the value, the additional tasks are scheduled into a queue and wait for the current tasks to be complete. Setting this parameter properly based on your bandwidth can improve the upload speed.
 - Download tasks concurrent number: Specify the maximum number of download tasks that can be performed at the same time. If the number of download tasks is larger than the value, the additional tasks are scheduled into a queue and wait for the current tasks to be complete. Setting this parameter properly based on your bandwidth can improve the upload speed.
 - overtime: Specify the timeout period for tasks.
 - uploadpart size: Specify the part size in multipart upload tasks. When the file to be uploaded is too large or the network condition is poor, you can set an appropriate part size to upload the object in multiple parts.
 - retry times: Specify the allowed retry times in upload or download tasks.
- Mail settings

You can click Settings to set your E-mail account. All operations related to mails in ossbrowser are performed by the account.

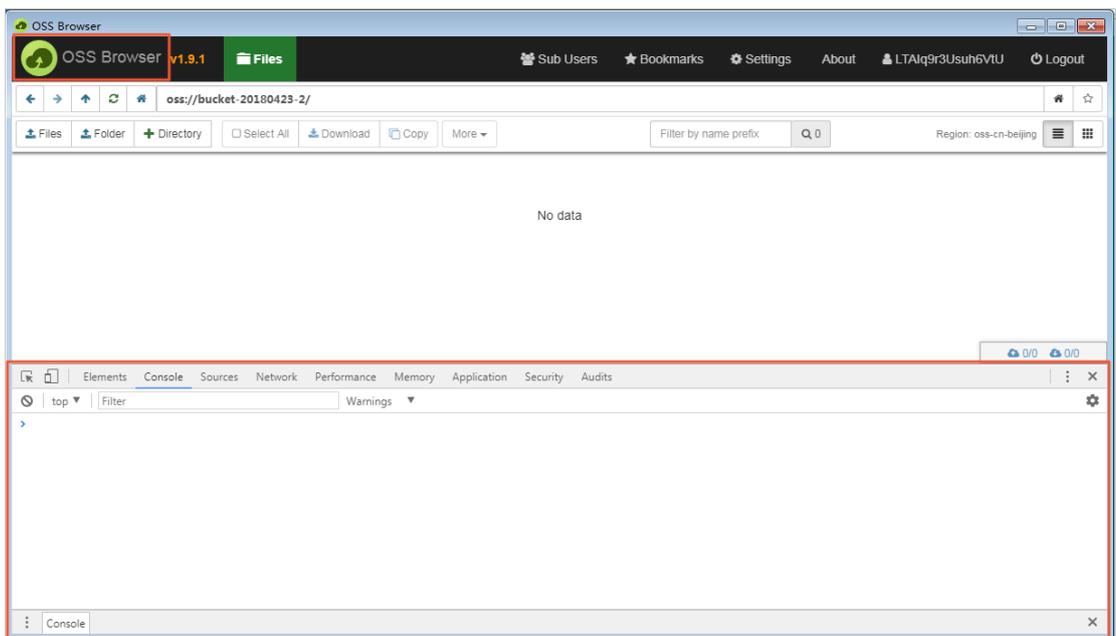
- Logging settings
 - Enable the debug mode.

You can enable the debug mode in the following two methods to view the logs generated by upload, download, and other operations.

■ Method 1: Click Settings, and then click Open debug.



■ Method 2: Continually click the OSS Browser logo at the upper left corner for 10 times.



- Enable logging.

You can select whether to enable the logging function in the Settings dialog box.

- Select Local log to enable the local logging function to collect error logs. Logs collected by ossbrowser are stored in the following paths by default:
 - Linux: `~/config/oss-browser/log.log`
 - Mac: `~/Library/Logs/oss-browser/log.log`
 - Windows: `%USERPROFILE%\AppData\Roaming\oss-browser\log.log`
- Select local info file log to collect normal local file information.

2.2 Permission management

Log on to ossbrowser as a RAM user

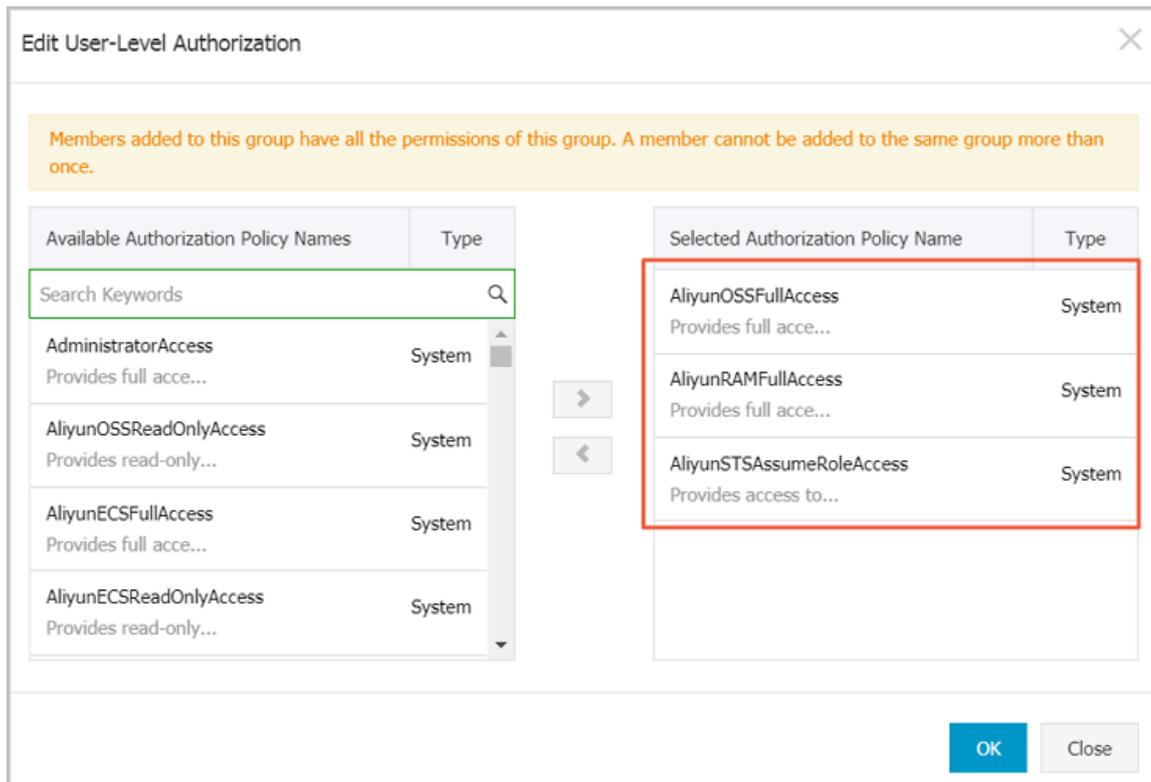
To ensure data security, we recommend that you log on to ossbrowser by using the AccessKey (AK) of a RAM user. To log on to ossbrowser, follow these steps:

1. Create a RAM user and an AK. For more information, see [Create a RAM user](#).

RAM users can be classified into two types based on their permissions:

- **Administrator RAM user:** Indicates a RAM user with administration permissions. For example, a RAM user that can manage all buckets and authorize other RAM users is an administrator RAM user. You can log on to the RAM console with your Alibaba Cloud account to create an administrator RAM

user and grant permissions to the administrator RAM user, as shown in the following figure.



- **Operator RAM user:** Indicates a RAM user that only has the read-only permission on a bucket or a directory. The administrator can [Grant permissions with a simple policy](#) to authorize a RAM user.



Note:

You can grant lower-level permissions to RAM users. For details, see [Access control](#).

2. Set the following parameters to log on to ossbrowser:

- **Endpoint:** Use the default value.
- **AccessKeyId and AccessKeySecret:** Enter the AK of the RAM user.
- **Preset OSS Path:**
 - **Administrator RAM users with administration permissions on all buckets:** No configuration is required.
 - **Operator RAM users:** Configurations are required. Enter the path of the OSS bucket or sub-directory that you want to access (the RAM user must have

permission to access the OSS bucket or sub-directory). The path format is as follows: oss:// bucket name/sub-directory name/.

- **Remember:** Select to save the AK. When you log on to ossbrowser later, you can simply click AK Histories and select the saved AK instead of entering the AK repeatedly. Do not select this option if you use a shared computer.

The screenshot shows the 'AK Login' form in the ossbrowser interface. At the top, there are two tabs: 'AK Login' (selected) and 'Token Login'. The form contains the following fields and controls:

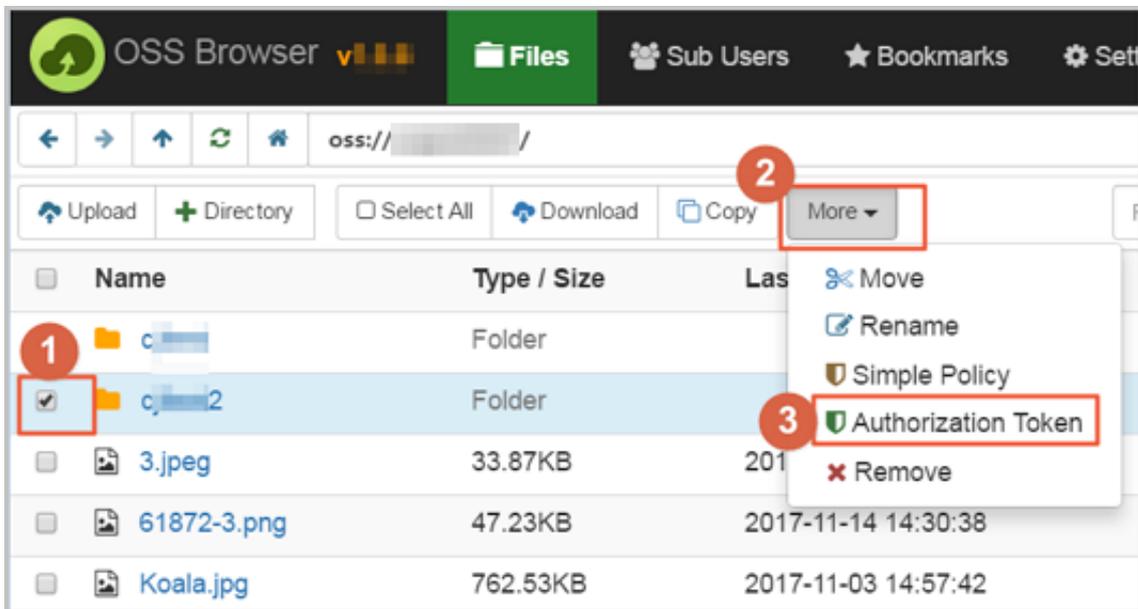
- Endpoint:** A dropdown menu set to 'Customize' and a text input field containing 'http://{region}.aliyuncs.com'.
- AccessKeyId:** A text input field with a masked value ending in 'm'.
- AccessKeySecret:** A text input field with a masked value consisting of dots.
- Preset OSS Path:** A text input field with a masked value, highlighted by a red rectangular box.
- Region:** A dropdown menu set to 'East China 1(Hangzhou)'.
- Description:** A text input field.
- Remember:** A checked checkbox with a 'Remember' label and a help icon.
- Buttons:** A blue 'Login' button and a grey 'AK Histories' button.

Log on to ossbrowser with a temporary authorization code

You can use a temporary authorization code to log on to ossbrowser. You can provide authorized users with a temporary authorization code to allow them to access a directory under your bucket temporarily before the authorization code expires. The temporary authorization code automatically becomes invalid after it expires.

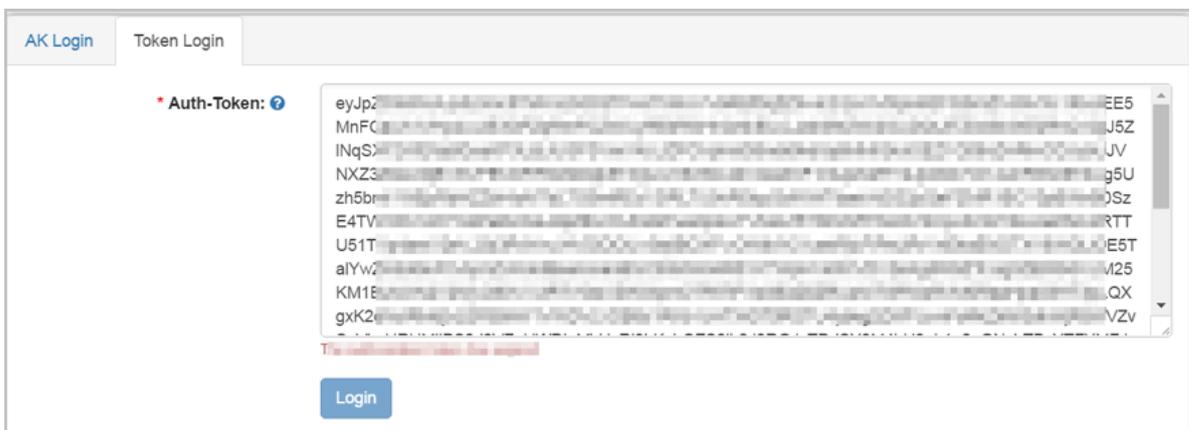
1. Generate a temporary authorization code.

Use the AK of an administrator RAM user to log on to ossbrowser. Select the object or directory to be accessed temporarily by the authorized users, and generate a temporary authorization code, as shown in the following figure.



2. Log on to ossbrowser with the authorization code.

The temporarily authorized users can use the authorization code to log on to ossbrowser before it expires, as shown in the following figure.



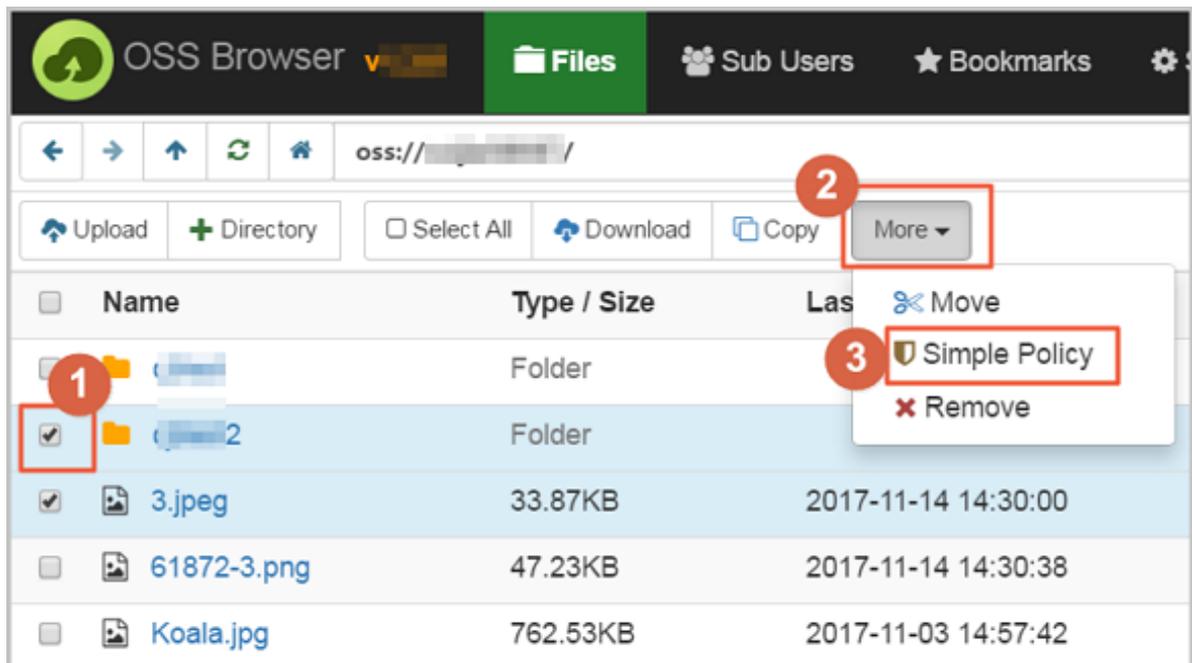
Grant permissions with a simple policy

After logging on to ossbrowser as an administrator RAM user, you can Grant permissions with a simple policy to create an operator RAM user, or grant an operator RAM user the read-only or read/write permission on a bucket or a directory.

 **Note:**

Alibaba Cloud ossbrowser provides simple policy authorization, which is an access control feature based on the Alibaba Cloud RAM service. You can also log on to the RAM console through the official website of Alibaba Cloud to manage your RAM user more precisely.

1. Select one or more objects or directories to be accessed temporarily by the authorized users and then click Simple Policy, as shown in the following figure.



2. On the Simplify policy authorization dialog box, select Privileges.
3. You can also grant permissions to an existing operator RAM user or create a new operator RAM user in this dialog box.

 Notice:

To use simple policy authorization, you must log on to ossbrowser by using the AK of an RAM user that has the RAM configuration permission, for example, the AK of an administrator RAM user that has the RAM configuration permission.



The policy is generated in text. You can view, copy, and use the text as needed. For example, you can copy the policy text and use it to edit the authorization rules for RAM users and roles in the RAM console.

3 ossutil

3.1 Overview

ossutil is a command-line tool that is used in Windows, Linux, or macOS to manage data in OSS. This tool provides a wide range of convenient and simple commands to manage objects and buckets.

ossutil allows you to perform the following operations:

- Bucket management, such as creating, listing, and deleting buckets.
- Object management, such as uploading, downloading, listing, copying, and deleting objects.
- Part management, such as listing and deleting parts.

Install ossutil

For more information about how to download and install ossutil, see [#unique_23](#).

Commonly used commands

The following table describes the commands that are supported in ossutil.

Command	Description
#unique_24	Uploads a local file to an appendable object in OSS.
#unique_25	Adds, modifies, queries, or deletes encryption configurations for a bucket.
#unique_26	Adds, modifies, queries, or deletes bucket policy configurations for a bucket.
#unique_27	Adds, modifies, queries, or deletes tagging configurations for a bucket.
#unique_28	Exports object content to ossutil.
<code>config</code>	Generates a configuration file to store OSS access information.
#unique_30	Adds, modifies, queries, or deletes cross-origin resource sharing (CORS) configurations for a bucket.
#unique_31	Tests whether a bucket allows a specified cross-origin request.

Command	Description
#unique_32	Uploads, downloads, or copies objects.
#unique_33	Creates a symbolic link (also known as a soft link).
#unique_34	Obtains the storage size of a specified bucket, object, or folder.
#unique_35	Obtains the total size of all parts in a bucket, and the size of each part that has not been uploaded.
#unique_36	Calculates the CRC64 or MD5 value of a local file.
#unique_37	Obtains help information about a command. We recommend that you use the <code>help</code> command to obtain information about a specified command.
#unique_38	Adds, modifies, queries, or deletes lifecycle configurations for a bucket.
#unique_39	Lists the parts that have not been uploaded for a specified object.
#unique_40	Adds, modifies, queries, or deletes logging configurations for a bucket.
#unique_41	Lists buckets, objects, or parts.
#unique_42	Creates a bucket.
#unique_43	Creates a directory in a bucket.
#unique_44	Adds, modifies, queries, or deletes tagging configurations for an object.
#unique_45	Monitors access to OSS, and troubleshoots problems caused during the upload and download process by network faults or incorrect parameter settings.
#unique_46	Reads the description of a symbolic link object.
#unique_47	Adds, modifies, queries, or deletes hotlink protection configurations for a bucket.
#unique_48	Restores a single object from the frozen state to the readable state.
#unique_49	Configures or queries pay-by-requester configurations for a bucket.
#unique_50	Deletes buckets, objects, or parts.
#unique_51	Configures the ACL for a bucket or object.
#unique_52	Configures the metadata of an uploaded object.

Command	Description
#unique_53	Generates signed URLs for third-party users to access objects in a bucket.
#unique_54	Obtains the description of a specified bucket or object.
#unique_55	Updates the ossutil version.
#unique_56	Adds, modifies, queries, or deletes static website hosting and back-to-origin configurations for a bucket.

3.2 Download and installation

This topic describes how to download and install ossutil.

Version and runtime environment

- Current version: 1.6.7
- Runtime environment
 - Windows/Linux/macOS
 - Supported architectures: x86 (32-bit and 64-bit)

Download URLs

- [Linux x86 32-bit](#)
- [Linux x86 64-bit](#)



Notice:

When copying the preceding URLs into the wget command to download ossutil, delete the `?spm = xxx` section from the URLs.

- [Windows x86 32-bit](#)
- [Windows x86 64-bit](#)
- [macOS x86 32-bit](#)
- [macOS x86 64-bit](#)

Quick installation

Download the package based on your operating system and run the corresponding binary file.

- Install ossutil in Linux (64-bit Linux system is used as an example)

1. Download the ossutil installation package:

```
wget http://gosspublic.alicdn.com/ossutil/1.6.7/ossutil64
```

2. Modify the file execution permissions:

```
chmod 755 ossutil64
```

3. Generate a configuration file by following the interactive processing:

```
./ossutil64 config
This command generates a configuration file to
store configuration information. Enter the path
of the configuration file. The default path is
/home/user/.ossutilconfig. If you press Enter
without specifying a 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 the path.
If the path of the configuration file is not
specified, the default path /home/user/.ossutilcon
fig is used.
The following parameters are ignored if you
press Enter without configuring them. For more
information about the parameters, run the help
config command.
Enter the endpoint: http://oss-cn-shenzhen.aliyuncs
.com
Enter the AccessKey ID: your AccessKey ID
Enter the AccessKey Secret: your AccessKey Secret
Enter the STS token:
```

- endpoint : specifies the domain name of the region to which the bucket belongs. For more information, see [Regions and endpoints](#).
- accessKeyId : For more information about how to view the AccessKey ID, see [#unique_14](#).
- accessKeySecret : For more information about how to view the AccessKey secret, see [#unique_14](#).
- stsToken : This option is required only when you use a temporary STS token to access the OSS bucket. Otherwise, you can leave this parameter unspecified. For more information about how to generate an STS token, see [Temporary access credential](#).



Note:

For more information about the configuration file, see [config](#).

- Install ossutil in Windows (64-bit Windows system is used as an example)
 1. Download the ossutil installation package.
 2. Decompress the package to a specified folder, and then double-click the `ossutil . bat` file.
 3. Generate the configuration file. For more information about the parameters, see the configuration parameters described in the preceding Linux section.

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

- Install ossutil in macOS (64-bit macOS system is used as an example)

1. Download the ossutil installation package.

```
curl -o ossutilmac 64 http :// gosspublic . alicdn . com /  
ossutil / 1 . 6 . 7 / ossutilmac 64
```

2. Modify the file execution permissions:

```
chmod 755 ossutilmac 64
```

3. Generate the configuration file. For more information about the parameters, see the configuration parameters described in the preceding Linux section.

```
./ ossutilmac 64  config
```

3.3 Common commands

3.3.1 appendfromfile

The `appendfrom file` command is used to upload local file content to an appendable object in OSS.



Note:

For more information about append upload, see [#unique_61](#).

Command syntax

```
./ ossutil  appendfrom  file  local_file  _name  oss :// bucket /  
object  [-- meta = meta - value ] [-- payer  requester ]
```

If the appendable object does not exist, you can use the `-- meta` option to configure the object metadata, such as `-- meta " X - Oss - Meta - Author : test "`. The `-- meta` option cannot be used to append content to an object that already exists.

**Notice:**

- You must specify the name for objects uploaded using append upload.
- You can only append content to objects created using append upload.
- You can use the [#unique_52](#) command to modify the metadata of an object.

Examples

Upload an object by using append upload.

```
./ ossutil  appendfrom  file  / file / test . txt  oss :// bucket1 /
test . txt
total  append   64 ( 100 . 00 %) byte , speed  is   0 . 00 ( KB /
s )
local  file  size  is   64 , the  object  new  size  is   64 ,
average  speed  is   0 . 34 ( KB / s )
```

Common options

The following table describes the options you can add to the `appendfrom file` command.

Option	Description
<code>-- meta</code>	Configures the metadata of a specified object. The <code>--meta</code> option cannot be used to append content to an object that already exists.
<code>-- maxupspeed</code>	Specifies the maximum upload speed. Unit: KByte/s. Default value: 0 (unlimited).
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code> . If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- loglevel</code>	Specifies the log level. The default value is <code>null</code> , indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> • <code>info</code>: generates prompt logs. • <code>debug</code>: generates detailed logs that contain HTTP requests and responses.
<code>-- payer</code>	Specifies the payer of the request. You can set this option to <code>requester</code> to enable the pay-by-requester mode.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 proxies are supported. An example of the URL is <code>http://120.79. **. **:3128</code> or <code>socks5://120.79. **. **:1080</code> .

Option	Description
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.2 bucket-encryption

The `bucket - encryption` command is used to add, modify, query, or delete encryption configurations for a bucket.



Note:

For more information about bucket encryption, see [#unique_64](#).

Command syntax

- Add or modify bucket encryption configurations

```
./ossutil bucket - encryption -- method put oss :// bucket
-- sse - algorithm algorithmN ame [-- kms - masterkey - id
keyid ]
```

- `-- sse - algorithm` : specifies the encryption method for the bucket. Valid values: KMS and AES256.
- `-- kms - masterkey - id` : specifies the CMK ID for encryption. When `-- sse - algorithm` is set to AES256, do not add the `-kms-masterkey-id` option. When the `--sse-algorithm` parameter is set to KMS, add this option as required.



Note:

The use of a specified CMK ID for encryption is under public preview. To enable this feature, contact [After-sales technical support](#).

If the bucket has no encryption configurations, you can run this command to add a specified encryption method for this bucket. If the bucket has encryption configurations, you can run this command to overwrite the existing encryption configurations.

- Obtain bucket encryption configurations

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

- Delete bucket encryption configurations

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

Examples

- Add or modify bucket encryption configurations

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

- Obtain bucket encryption configurations

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

- Delete bucket encryption configurations

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

Common options

The following table describes the options you can add to the `bucket - encryption` command.

Option	Description
<code>-- sse - algorithm</code>	Specifies the server-side encryption algorithm. Valid values: KMS and AES256.
<code>-- kms - masterkey - id</code>	Specifies the CMK ID for encryption. When <code>-- sse - algorithm</code> is set to AES256, do not add the <code>-kms-masterkey-id</code> option. When the <code>--sse-algorithm</code> parameter is set to KMS, add this option as required.
<code>-- method</code>	Specifies the HTTP request method. Valid values: <ul style="list-style-type: none"> • <code>put</code>: adds or modifies bucket encryption configurations. • <code>get</code>: obtains bucket encryption configurations. • <code>delete</code>: deletes bucket encryption configurations.

Option	Description
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · <code>info</code>: generates prompt logs. · <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.**.**:3128</code> or <code>socks5://120.79.**.**:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.3 bucket-policy

The `bucket - policy` command is used to add, modify, query, or delete bucket policy configurations for a bucket.



Note:

For more information about bucket policies, see [#unique_66](#).

Command syntax

- Add or modify bucket policy configurations

```
./ ossutil bucket - policy -- method put oss :// bucket
local_json _file [ options ]
```

ossutil reads the `local_json _file` configuration file. If the bucket has no bucket policy configurations, ossutil writes the configurations to this configuration file. If the bucket has bucket policy configurations, new configurations will overwrite the existing configurations.



Note:

The `local_json _file` configuration file is in the JSON format as follows:

```
{
```

```

    " Version ": " 1 ",
    " Statement ": [
      {
        " Effect ": " Allow ",
        " Action ": [
          " ram : ListObject s "
        ],
        " Principal ": [
          " 1234567 "
        ],
        " Resource ": [
          "*"
        ],
        " Condition ": {}
      }
    ]
  }
}

```

- Obtain bucket policy configurations

```

./ ossutil bucket - policy -- method get oss :// bucket [
local_json _file ] [ options ]

```

The `local_json _file` parameter specifies the name of the configuration file. If this parameter is specified, bucket policy configurations will be saved as a local file. If this parameter is not specified, ossutil displays the bucket policy configurations.

- Delete bucket policy configurations

```

./ ossutil bucket - policy -- method delete oss :// bucket [
options ]

```

Examples

- Add bucket policies to allow anonymous users with specified IP addresses to access all resources in a bucket

```

./ ossutil bucket - policy -- method put oss :// bucket1 /
file / policy . json

```

The content of the `policy . json` configuration file is as follows:

```

{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " oss :*"
      ],
      " Effect ": " Allow ",
      " Principal ": [
        "*"
      ],
      " Resource ": [
        " acs : oss :* : 1746495857 60 **** : bucket1 ",

```

```

        "acs : oss :* : 1746495857 60 ****: bucket1 /*"
    ],
    "Condition ": {
        "IpAddress ": {
            "acs : SourceIp ": [
                " 10 . 10 . 10 . 10 "
            ]
        }
    }
}
]
}
}

```

- **Obtain bucket policy configurations**

```

./ ossutil . exe  bucket - policy  -- method  get  oss ://
bucket1
{
    " Version ": " 1 ",
    " Statement ": [
        {
            " Action ": [
                " oss :*"
            ],
            " Effect ": " Allow ",
            " Principal ": [
                "*"
            ],
            " Resource ": [
                " acs : oss :* : 1746495857 60 ****: bucket1 ",
                " acs : oss :* : 1746495857 60 ****: bucket1 /*"
            ],
            " Condition ": {
                " IpAddress ": {
                    " acs : SourceIp ": [
                        " 10 . 10 . 10 . 10 "
                    ]
                }
            }
        }
    ]
}
}

```

- **Delete bucket policy configurations**

```

./ ossutil . exe  bucket - policy  -- method  delete  oss ://
bucket1

```

Common options

The following table describes the options you can add to the `bucket - policy` command.

Option	Description
-- method	Specifies the HTTP request method. Valid values: <ul style="list-style-type: none"> · put: adds or modifies bucket policy configurations. · get: obtains bucket policy configurations. · delete: deletes bucket policy configurations.
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is https://120.79.***:3128 or socks5://120.79.***:1080.
-- proxy - user	Specifies the username of the proxy server. The default value is null.
-- proxy - pwd	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.4 bucket-tagging

The `bucket - tagging` command is used to add, modify, query, or delete tagging configurations for a bucket.



Note:

For more information, see [Bucket tagging](#).

Command syntax

- Add or modify bucket tagging configurations

```
./ ossutil bucket - tagging -- method put oss :// bucket
tagkey1 # tagvalue1 tagkey2 # tagvalue2
```

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



Note:

- Only the bucket owner and authorized users can configure bucket tagging. Otherwise, OSS returns 403 Forbidden with error code AccessDenied.
- Separate the key from the value in each tag with number signs (#).
- You can configure a maximum of 20 tags for a bucket. Separate multiple tags with spaces.
- The key and value of the tag can contain letters, digits, spaces, and special characters such as
+ - = . _ : /
- The tag key is required. The tag key can be a maximum of 64 Bytes in length and cannot start with `http ://`, `https ://`, or `Aliyun` .
- The tag value is optional. The tag value can be a maximum of 128 Bytes in length.
- The key and value of the tag must be encoded in UTF-8.

- Query bucket tagging configurations

```
./ ossutil bucket - tagging -- method get oss :// bucket
```

- Delete bucket tagging configurations

```
./ ossutil bucket - tagging -- method delete oss :// bucket
```

Examples

- Add bucket tagging configurations

```
./ ossutil bucket - tagging -- method put oss :// bucket1  
tag1 # test1 tag2 # test2
```

- Query bucket tagging configurations

```
./ ossutil bucket - tagging -- method get oss :// bucket1
```

- Delete bucket tagging configurations

```
./ ossutil bucket - tagging -- method delete oss :// bucket1
```

Common options

The following table describes the options you can add to the `bucket - tagging` command.

Option	Description
<code>-- method</code>	Specifies the HTTP request method. Valid values: <ul style="list-style-type: none"> · <code>put</code>: adds or modifies bucket tagging configurations. · <code>get</code>: obtains bucket tagging configurations. · <code>delete</code>: deletes bucket tagging configurations.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · <code>info</code>: generates prompt logs. · <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***.3128</code> or <code>socks5://120.79. **. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.5 cat

The `cat` command is used to export the content of a specified object to ossutil.

Command syntax

```
./ ossutil cat oss :// bucket / object [-- payer requester ]
```



Note:

We recommend that you only use this command to view the content of a TXT file.

Examples

View the content of a specified object.

```
./ ossutil cat oss :// bucket1 / test . txt
language = ch
endpoint = oss - cn - hangzhou . aliyuncs . com
```

```
0 . 273027 ( s ) elapsed
```

Common options

The following table describes the options you can add to the `cat` command.

Option	Description
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code> . If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- loglevel</code>	Specifies the log level. The default value is <code>null</code> , indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> <code>info</code>: generates prompt logs. <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- payer</code>	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to <code>requester</code> .
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79. **: **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is <code>null</code> .
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is <code>null</code> .



Note:

For more information about common options, see [#unique_62](#).

3.3.6 config

The `config` command is used to create a configuration file to store OSS access information. You can add the `- c` option to other commands to provide OSS access information.

Command syntax

```
./ ossutil config [- e endpoint ] [- i id ] [- k key ] [- t token ] [- L language ] [-- output - dir outdir ] [- c file ]
```



Note:

This command can be used in both interactive and non-interactive modes. We recommend that you run the command in interactive mode to ensure security.

Examples

- Generate a configuration file in interactive mode

```
./ ossutil  config
This command generates a configuration file . Enter
the path of the configuration file . The default
path is / home / user /. ossutilconfig . If you press
Enter without specifying a different destination ,
the file is generated in the default path . If you
want to generate the file in another path , set
the -- config - file option to the path .
If no configuration file path is entered , the
default configuration file path / home / user /.
ossutilconfig is used .
The following parameters are ignored if you press
Enter without configuring them . For more informatio
n about the parameters , run the help config command
.
Enter the endpoint : http :// oss - cn - shenzhen . aliyuncs .
com
Enter the AccessKey ID : yourAccess KeyID
Enter the AccessKey secret : yourAccess KeySecret
Enter the STS token :
```

- endpoint : specifies the domain name of the region to which the bucket belongs. For more information, see [Regions and endpoints](#).
 - accessKeyID : For more information about how to view the AccessKey ID, see [#unique_14](#).
 - accessKeySecret : For more information about how to view the AccessKey secret, see [#unique_14](#).
 - stsToken : This option is required only when you use a temporary STS token to access the OSS bucket. Otherwise, you can leave this parameter unspecified. For more information about how to generate an STS token, see [Temporary access credential](#).
- Generate a configuration file in non-interactive mode

```
./ ossutil  config - e oss - cn - beijing . aliyuncs . com - i
LTAIbZcdVC mQ **** - k D26oqKBudx DRBg8Wuh2E WDBrM0 **** -
L CH - c / myconfig
```

If you run the config command with options other than -- language and -- config - file , the non-interactive mode is used. All configuration items are specified using options.

Configuration file format

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

```
[ Credentials ]
    language = CH
    endpoint = oss . aliyuncs . com
    accessKeyID = your_key_id
    accessKeySecret = your_key_secret
    stsToken = your_sts_token
    outputDir = your_output_dir
[ Bucket - Endpoint ]
    bucket1 = endpoint1
    bucket2 = endpoint2
    ...
[ Bucket - Cname ]
    bucket1 = cname1
    bucket2 = cname2
    ...
[ AkService ]
    ecsAk = http://100.100.100.200/latest/meta-data/
    /Ram/security-credentials/EcsRamRoleTesting
```

- **Bucket-Endpoint:** specifies an individual endpoint for each specified bucket. If you configure the Bucket-Endpoint option, ossutil searches for the specified endpoint when you perform operations on the bucket. If the specified endpoint exists, ossutil manages the bucket through that endpoint. Otherwise, ossutil manages the bucket through the endpoint specified in the Credentials option.
- **Bucket-Cname:** specifies an individual origin domain for each specified bucket. If you configure the Bucket-Cname option, ossutil searches for the specified origin domain when you perform operations on the bucket. If the specified origin domain exists, ossutil replaces the endpoints specified in the Bucket-Endpoint and Credentials options with the origin domain. For more information about origin domains, see [Configure CNAME](#).
- **AkService:** This option is not added by default. This option is required if you need to use a RAM role bound to an ECS instance to perform operations on OSS. When you configure this option, you need only to set EcsRamRoleTesting to the RAM role bound to the ECS instance. After configuring the AkService option, you do not need to configure the accessKeyID, accessKeySecret, and stsToken options. If these options are configured, the configurations of these options instead of the AkService option take effect. For more information about how to bind a RAM role to an ECS instance, see [Use RAM roles in the console](#). For more information about how to bind a RAM role to an ECS instance, see [#unique_71](#).

**Notice:**

- In the new version of ossutil, the Bucket-Endpoint and Bucket-Cname options do not need to be specified in interactive mode. However, the two options still take effect in the configuration file. You can specify an endpoint or origin domain for each bucket in the configuration file.
- ossutil can specify endpoints for multiple regions. The priorities of all endpoint configurations are as follows: endpoint specified by the `-- endpoint` option in commands > endpoint specified in Bucket-Cname > endpoint specified in Bucket-Endpoint > endpoint specified in Credentials. If you specify the `-- endpoint` (abbreviated as `- e`) option when running a command, the value of the `-- endpoint` option takes precedence. If this option is not specified, ossutil searches for the endpoints specified in Bucket-Cname, Bucket-Endpoint, and Credentials in the configuration file. The endpoint with the highest priority is used.

Common options

The following table describes the options you can add to the `config` command to generate the specified configuration items.

Option	Description
<code>- e , -- endpoint</code>	Specifies the endpoint in the [Credentials] section of the configuration file.
<code>- i , -- access - key - id</code>	Specifies the AccessKey ID in the [Credentials] section of the configuration file.
<code>- k , -- access - key - secret</code>	Specifies the AccessKey secret in the [Credentials] section of the configuration file.
<code>- t , -- sts - token</code>	Optional. This option specifies the STS token in the [Credentials] section of the configuration file.
<code>-- output - dir</code>	Specifies the directory in which output objects are located. Output objects include report objects generated due to errors that occur when you use the <code>cp</code> command to copy multiple objects. The default value is the <code>ossutil_ou tput</code> directory in the current directory.

Option	Description
- L , -- language	Specifies the language ossutil uses. Valid values: CH and EN. Default value: CH. If you plan to set this option to CH, ensure that your system supports UTF-8 encoding.
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.



Note:

For more information about common options, see [#unique_62](#).

3.3.7 cors

The `cors` command is used to add, modify, query, or delete the CORS configurations for a bucket.



Note:

For more information about CORS, see [#unique_73](#).

Command syntax

- Add or modify CORS configurations for a specified bucket

```
./ossutil cors --method put oss://bucket local_xml_file
```

ossutil reads the `local_xml_file` configuration file. If the bucket has no CORS rules configured, ossutil adds the corresponding CORS rules to this configuration file. If the bucket has CORS rules configured, the new rules overwrite the existing rules.



Note:

The `local_xml_file` configuration file is in the XML format as follows:

```
<? xml version = " 1 . 0 " encoding = " UTF - 8 "? >
  < CORSConfig uration >
    < CORSRule >
      < AllowedOri gin > www . aliyun . com </ AllowedOri gin
    >
      < AllowedMet hod > PUT </ AllowedMet hod >
      < MaxAgeSeco nds > 10000 </ MaxAgeSeco nds >
    </ CORSRule >
```

```
</ CORSConfig uration >
```

- Obtain CORS configurations for a bucket

```
./ ossutil cors -- method get oss :// bucket [ local_xml_
file ]
```

The `local_xml_ file` parameter specifies the name of the configuration file. If this parameter is specified, ossutil saves the obtained CORS configurations as a local file. If this parameter is not specified, ossutil displays the obtained CORS configurations.

- Delete CORS configurations for a bucket

```
./ ossutil cors -- method delete oss :// bucket
```

Examples

- Add CORS configurations for a bucket

```
./ ossutil cors -- method put oss :// bucket1 / file / cors
.xml
```

- Obtain CORS configurations for a bucket

```
./ ossutil cors -- method get oss :// bucket1 / file / cors
.xml
```

- Delete CORS configurations for a bucket

```
./ ossutil cors -- method delete oss :// bucket1
```

Common options

The following table describes the options you can add to the `cors` command.

Option	Description
<code>-- method</code>	Specifies the HTTP request method. Valid values: <ul style="list-style-type: none"> • <code>put</code>: adds or modifies CORS configurations. • <code>get</code>: obtains CORS configurations. • <code>delete</code>: deletes CORS configurations.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> • <code>info</code>: generates prompt logs. • <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.

Option	Description
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.**.**:3128</code> or <code>socks5://120.79.**. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.8 cors-options

The `cors - options` command is used to test whether a bucket has cross-origin resource sharing (CORS) configured.



Note:

- For more information about CORS, see [#unique_73](#).
- For more information about commands used to configure CORS, see [#unique_30](#).

Command syntax

```
./ ossutil  cors - options  -- acr - method < value > -- origin <
value > -- acr - headers < value > oss :// bucket /[ object ]
```

The `cors - options` command is used to send an HTTP OPTIONS request to OSS to test whether a cross-origin request is allowed.

- `-- acr - method` : specifies the value of the Access-Control-Request-Method field in the HTTP header. Valid values: GET, PUT, POST, DELETE, and HEAD.
- `-- origin` : specifies the value of the Origin field in the HTTP header. This parameter specifies the origin where a cross-origin request is from and is used to identify the origin that you will allow to access your bucket.
- `-- acr - headers` : specifies the value of the Access-Control-Request-Headers field in the HTTP header. This parameter specifies actual headers except commonly used headers. To specify multiple headers, separate different headers with commas (,) and enclose the headers with double quotation marks (").
Example: `--acr-headers "header1,header2,header3."`

Examples

```
./ossutil cors - options -- acr - method put -- origin " www
. aliyun . com " oss :// bucket1
Access - Control - Allow - Methods : GET , POST , PUT
Access - Control - Allow - Origin : *
Access - Control - Max - Age : 0
```

Common options

The following table describes the options you can add to the `cors - options` command.

Option	Description
<code>-- origin</code>	Specifies the value of the Origin field in the HTTP header. This parameter specifies the origin where a cross-origin request is from and is used to identify the origin that you will allow to access your bucket.
<code>-- acr - method</code>	Specifies the value of the Access-Control-Request-Method field in the HTTP header. Valid values: GET, PUT, POST, DELETE, and HEAD.
<code>-- acr - headers</code>	Specifies the value of the Access-Control-Request-Headers field in the HTTP header. This parameter specifies actual headers except commonly used headers. To specify multiple headers, separate different headers with commas (,) and enclose the headers with double quotation marks ("). Example: <code>--acr-headers "header1,header2,header3."</code>
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> <code>info</code>: generates prompt logs. <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.**.**:3128</code> or <code>socks5://120.79.**. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.

Option	Description
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.9 cp

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

Command syntax

- Upload objects

```
./ ossutil cp file_url cloud_url [- r ] [- f ] [- u ]  
  [-- output - dir = odir ] [-- bigfile - threshold = size ] [--  
  checkpoint - dir = cdir ] [-- snapshot - path = sdir ] [-- payer  
  requester ]
```

- Download objects

```
./ ossutil cp cloud_url file_url [- r ] [- f ] [- u ]  
  [-- output - dir = odir ] [-- bigfile - threshold = size ] [--
```

```
checkpoint - dir = cdir ] [-- range = x - y ] [-- payer requester ]
```

- Copy objects

```
./ ossutil cp cloud_url cloud_url [- r ] [- f ] [- u ]
[-- output - dir = odir ] [-- bigfile - threshold = size ] [--
checkpoint - dir = cdir ] [-- payer requester ]
```

Examples

- Upload objects

- Upload a single object

```
./ ossutil cp a . txt oss :// bucket / path
```

- Upload a single object and specify the `-- meta` option

When uploading an object, you can use the `-- meta` option to configure object metadata in the `header : value # header : value ...` format. Run the following command to upload the `a . txt` file and configure its metadata:

```
./ ossutil cp a . txt oss :// bucket / path -- meta = Cache
- Control : no - cache # Content - Encoding : gzip
```



Note:

For more information about meta settings, see [#unique_52](#).

- Upload a folder

You can add the `-r` option to the `cp` command to upload a folder to OSS. The command is as follows:

```
./ ossutil cp - r dir oss :// bucket / path
```



Note:

If the object to be uploaded is a symbolic link that maps to a local folder, add a forward slash (/) to the symbolic link when you use the `cp` command to upload the object.

```
./ ossutil cp -r symbolic_link / oss :// bucket / path
```

- Upload an object or folder and configure the maximum upload speed

When uploading an object, use the `-- maxupspeed` option to configure the maximum upload speed measured in KByte/s. The default value is 0, which indicates that there is no maximum upload speed.

■ Upload an object and set the maximum upload speed to 1 MByte/s

```
./ ossutil cp a .jpg oss :// bucket / path --
maxupspeed 1024
```

■ Upload a folder and set the maximum upload speed to 1 MByte/s

```
./ ossutil cp -r dir oss :// bucket / path --
maxupspeed 1024
```

- Upload multiple objects that meet specified conditions

When copying multiple objects, you can add the `-- include` / `-- exclude` option to the `cp` command to specify conditions for which to select objects.

The `-- include` / `-- exclude` option supports the following formats:

- `*`: matches all characters. For example, `*. txt` represents any `txt` objects.
- `?`: matches any single character. For example, `abc ?. jpg` represents all JPG objects whose names contain `abc` followed by any single character, such as `abc1.jpg`.
- `[sequence]`: matches any characters in a sequence. For example, `abc [1 - 5]. jpg` specifies the objects whose names begin with `abc` and are followed by a number contained in the sequence `[1-5]`, such as `abc1 . jpg ~ abc5 . jpg`.
- `[! sequence]`: matches any characters that are not in a sequence. For example, `abc [! 0 - 7]. jpg` specifies the objects whose names begin with `abc` and are not followed by a number contained in the sequence `[0-7]`, such as `abc0 . jpg ~ abc7 . jpg`.



Note:

- Names that include directories, such as `-- include "/usr/test/.jpg"`, are not supported.
- To specify the `-- include /-- exclude` option, you must also specify the `-- recursive (-r)` option.
- One rule can contain multiple conditions specified by `include` and `exclude`. If these conditions are set, `ossutil` reads each rule sequentially from left to right to obtain the final matching results.

If the `test.txt` object exists in a valid folder, 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 in the TXT format and is excluded. The final matching results exclude the `test.txt` object.
- **Rule 2:** `-- exclude "*.txt" -- include "*test*"`. When `ossutil` reads the `-- exclude "*.txt"` condition, the matching results exclude the `test.txt` object. When `ossutil` reads the `-- include "*test*"` condition, the `test.txt` object name contains `test` and matches the condition. 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 in the TXT format and is excluded. 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.

Examples:

- Upload all objects that are in the `txt` format.

```
./ ossutil cp dir / oss :// my - bucket / path -- include
"* . txt " - r
```

- Upload all objects that contain `abc` in their names and are not in the `jpg` or `txt` format.

```
./ ossutil cp dir / oss :// my - bucket / path -- include
"* abc *" -- exclude "*. jpg " -- exclude "*. txt " - r
```

- Upload an object and modify its storage class

When uploading an object, you can use the `-- meta` option to modify the storage class of the object.

- Upload an object and set its storage class to IA.

```
./ ossutil cp dir / sys . log oss :// my - bucket / path
-- meta X - oss - Storage - Class : IA
```

- Upload a folder and set the storage class of all objects in the folder to IA.

```
./ ossutil cp dir / oss :// my - bucket / path -- meta
X - oss - Storage - Class : IA - r
```

- Upload an object and specify the server-side encryption method

You can specify the server-side encryption method when you upload an object. After you encrypt the object, you can store the object in a bucket. For more information about server-side encryption, see [#unique_64](#).

- Upload an object and set the server-side encryption method to AES256

```
./ ossutil cp a . txt oss :// my - bucket / path -- meta
= x - oss - server - side - encryption : AES256
```

- Upload an object and set the server-side encryption method to KMS

```
./ ossutil cp a . txt oss :// my - bucket / path -- meta
= x - oss - server - side - encryption : KMS
```

When you use KMS to encrypt an object, OSS creates a master key in the KMS console for the object, which will incur a small fee when the KMS API operation is called.

- Upload a folder without uploading the existing objects

If you specify the `-- update` (abbreviated as `- u`) option when uploading multiple objects, ossutil only uploads the objects that do not exist at the destination or that have a later last modification time than that of the destination objects. The command is as follows:

```
./ ossutil cp - r dir oss :// bucket1 / path - u
```

This option can be used to perform incremental upload on objects that have not been successfully uploaded.

- Upload a folder and generate snapshot information

If you specify the `-- snapshot - path` option when uploading multiple objects, ossutil takes a snapshot of the upload and stores the snapshot information in the specified directory. When this option is specified the next time objects are uploaded, ossutil will read the snapshot information from the specified directory to perform incremental upload. The command is as follows:

```
./ ossutil cp - r dir oss :// bucket1 / path -- snapshot -  
path = path
```



Notice:

- The `-- snapshot - path` option is used in certain scenarios to accelerate the incremental upload or download of multiple objects. This option can be used when the number of objects is large and no other users modify the corresponding objects in OSS during two uploads. This option cannot be used to copy objects.
- The `-- snapshot - path` option records the local `lastModifiedTime` of uploaded or downloaded objects, and then compares that recorded `lastModifiedTime` with that of objects to be uploaded or downloaded next time to determine which objects can be skipped. When using this option, make sure that the corresponding objects in OSS are not modified during the two uploads or downloads. In other scenarios where objects are updated in OSS during 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`. If the snapshot information is not necessary to retain, remove the directory.
- Additional overheads are required to read and write snapshot information. We recommend that you do not use this option in any of the following scenarios: The number of objects to be uploaded or downloaded is small. The network is properly connected. Other users need to perform operations on those objects. In this case, you can use the `-- update` option to perform incremental upload or download.
- The `-- update` and `-- snapshot - path` options can be used together. ossutil first uses the `--snapshot-path` option to determine whether to skip the upload or download of an object. If the upload or download is not skipped by `--snapshot-path`, ossutil then uses the `-- update` option to determine whether to skip the upload or download of the object.

- Upload an object to a bucket that has the pay-by-requester mode enabled

```
./ ossutil cp dir / test . mp4 oss :// payer / -- payer =
requester
```

- Download objects

- Download a single object

```
./ ossutil cp oss :// my - bucket / path / test1 . txt / dir
```

- Download parts of an object within a specified range

You can use the `-- range` option to download parts of an object within a specified range.

For example, download the 10th to 20th characters from the `test1 . txt` object to your local device as a file.

```
./ ossutil cp oss :// my - bucket / path / test1 . txt / dir
-- range = 10 - 20
Succeed : Total num : 1 , size : 11 . OK num : 1 (
download 1 objects ).
```

```
0 . 290769 ( s ) elapsed
```

- Download an object from a bucket that has the pay-by-requester mode enabled

```
./ ossutil cp oss :// payer / test . mp4 dir / -- payer = requester
```

- Download a folder

```
./ ossutil cp - r oss :// my - bucket / path / dir
```

- Download a folder and specify the -- update option

If you specify the -- update option when downloading multiple objects, ossutil only downloads the objects that do not exist at the destination, or that have a later last modification time than that of the destination objects. The command is as follows:

```
./ ossutil cp - r oss :// bucket / path / dir -- update
```

This option can be used to perform incremental download on objects that have not been successfully downloaded.

- Download a folder and generate snapshot information

If you specify the -- snapshot - path option when downloading multiple objects, ossutil takes a snapshot of the download and stores the snapshot information in the specified directory. When this option is specified the next time objects are downloaded, ossutil will read the snapshot information

from the specified directory to perform incremental download. For more information, see [--snapshot-path](#). The command is as follows:

```
./ ossutil cp -r oss :// bucket1 / path dir -- snapshot -
path = path
```

- Download multiple objects that meet specified conditions

When downloading multiple objects, you can use `-- include` / `-- exclude` to specify conditions for which to select objects. For more information, see [cp](#).

■ Download all objects that are not in the *jpg* format

```
./ ossutil cp oss :// my - bucket / path dir / -- exclude
"*. jpg " - r
```

■ Download all objects that contain *abc* in their names and are not in the *jpg* or *txt* format

```
./ ossutil cp oss :// my - bucket1 / path dir / --
include "* abc *" -- exclude "*. jpg " -- exclude "*. txt "
- r
```

· Copy objects

Only objects can be copied. Parts cannot be copied. You cannot copy objects across regions.

- Copy a single object

```
./ ossutil cp oss :// bucket / path1 / a oss :// bucket /
path2 /
```

- Copy and rename a single object

```
./ ossutil cp oss :// bucket / path1 / a oss :// bucket /
path2 / b
```

- Copy multiple objects that meet specified conditions

When copying multiple objects, you can use the `-- include` / `-- exclude` command to specify conditions for which to select objects. For more information, see [cp](#).

■ Copy all objects that are not in the `jpg` format

```
./ ossutil cp oss :// my - bucket1 / path oss :// my -  
bucket2 / path -- exclude "*.jpg" - r
```

■ Copy all objects that contain `abc` in their names and are not in the `jpg` or `txt` format

```
./ ossutil cp oss :// my - bucket1 / path oss :// my -  
bucket2 / path -- include "* abc *" -- exclude "*.jpg" --  
exclude "*.txt" - r
```

- Copy an object and modify its storage class**■ Set the storage class of an existing object to Archive**

```
./ ossutil cp oss :// my - bucket / path / 0104_6 .jpg  
oss :// my - bucket / path / 0104_6 .jpg -- meta X - oss -  
Storage - Class : Archive
```

■ Set the storage class of all files in an existing folder to Standard

```
./ ossutil cp oss :// my - bucket / path / oss :// my -  
bucket / path / -- meta X - oss - Storage - Class : Standard  
- r
```

**Notice:**

- You cannot use the `cp` command to convert the storage class of an object from Archive to any other storage class. You must first use the [#unique_48](#) command to restore the object to the readable state, and then use the `cp` command to modify its storage class.
- When you run the `cp` command to overwrite an object, it is considered overwritten and you may incur fees. If objects of the IA or Archive storage

class are overwritten 30 or 60 days after they are created, early deletion fees will be incurred. For more information, see [Billing items](#).

- Copy an object and specify the server-side encryption method

You can specify the server-side encryption method when you copy an object. After you encrypt the object, you can store the object in a bucket. For more information about server-side encryption, see [#unique_64](#).

- Copy an object and set the server-side encryption method to AES256

```
./ ossutil cp oss :// bucket / path1 / a oss :// bucket / path2 / -- meta = x - oss - server - side - encryption : AES256
```

- Copy an object and set the server-side encryption method to KMS

```
./ ossutil cp oss :// bucket / path1 / a oss :// bucket / path2 / -- meta = x - oss - server - side - encryption : KMS
```

When you use KMS to encrypt an object, OSS creates a master key in the KMS console for the object, which will incur a small fee when the KMS API operation is called.

- Copy a single object and specify the `-- meta` option

When copying an object, you can use the `-- meta` option to configure object metadata in the `header : value # header : value ...` format.

```
./ ossutil cp oss :// bucket / path1 / a oss :// bucket / path2 / -- meta = Cache - Control : no - cache
```

- Copy multiple objects between different buckets within the same region

You can add the `- r` option to the `cp` command to copy multiple objects between different buckets within the same region.

```
./ ossutil cp oss :// your_src_bucket / path1 / oss :// your_dest_bucket / path2 / - r
```

- Perform incremental copy on objects between different buckets within the same region

If you specify the `-- update` option when copying multiple objects, `ossutil` only copies the objects that do not exist at the destination or that have a later

last modification time than that of the destination objects. The command is as follows:

```
./ ossutil cp oss :// your_src_bucket / path1 / oss ://
your_dest_bucket / path2 / -r -- update
```

This option can be used to perform incremental copy on objects that have not been successfully copied.

- Copy an object from a bucket that has the pay-by-requester mode enabled to a common bucket

```
./ ossutil cp oss :// payer / test . mp4 oss :// my - bucket
/ path -- payer = requester
```

- Copy an object from a common bucket to a bucket that has the pay-by-requester mode enabled

```
./ ossutil cp oss :// my - bucket / path / test . mp4 oss
:// payer -- payer = requester
```

- Optimize the object upload, download, and copy performance

You can add the `--jobs` and `--parallel` options to the `cp` command to specify the number of concurrent operations. If the default number of concurrent operations set by `ossutil` does not meet your performance requirements, you can modify the values of these two options to adjust the performance.

- The `-- jobs` option specifies the number of concurrent tasks when multiple files are uploaded, downloaded, or copied.
- The `-- parallel` option specifies the number of concurrent operations performed on a large object during multipart upload, download, or copy.

`ossutil` calculates the number of concurrent operations based on the object size by default. This option does not work for small objects. The threshold for large objects to be uploaded, downloaded, or copied in multipart mode can be specified through the `-- bigfile - threshold` option. When you upload, download, or copy multiple objects, the actual number of concurrent operations is calculated by multiplying the number of jobs by the number of concurrent operations.



Warning:

- We recommend that you adjust the number of concurrent operations to a value smaller than 100 if the resources of your ECS instance or server (such

as network bandwidth, memory, and CPU) are limited. If resources such as the network, memory, and CPU are not fully occupied, you can increase the number of concurrent operations.

- If there are too many concurrent operations, the upload, download, and copy performance of ossutil may degrade or an EOF error may occur due to inter-thread resource switching and snatching. To resolve this issue, you must adjust the values of the `-- jobs` and `-- parallel` options as required. To perform pressure testing, set the two options to small values first, and slowly adjust them to the optimal values.

Common options

The following table describes the options you can add to the `cp` command.

Option	Description
<code>- r , -- recursive</code>	Recursively performs operations on objects in a bucket. If this option is specified, commands that support this option will perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, the commands will only perform operations on a single specified object.
<code>- f , -- force</code>	Forces an operation without prompting the user for confirmation.
<code>- u , -- update</code>	ossutil only uploads, downloads, or copies the objects that do not exist at the destination or that have a later last modification time than that of the destination objects.
<code>-- output - dir</code>	Specifies the directory in which output objects are located. Output objects include report objects generated due to errors that occur when you use the <code>cp</code> command to copy multiple objects. The default value is the <code>ossutil_output</code> directory in the current directory.
<code>-- bigfile - threshold</code>	Specifies the object size for which to start resumable data transfer. Unit: Byte. Valid values: non-negative integers. Default value: 100 MB.
<code>-- part - size</code>	Specifies the size of each part during multipart upload , download, or copy of large objects. Unit: Byte. ossutil calculates the appropriate part size based on the object size by default. You can set this option to any positive integer if you need to optimize performance or are operating under special constraints.

Option	Description
-- checkpoint - dir	Specifies the checkpoint directory. Default value: .ossutil_checkpoint. When a resumable data transfer fails, ossutil automatically creates a directory and records the checkpoint information in the directory. When the resumable transfer succeeds, ossutil deletes this directory. If this option is specified, ensure that the specified directory can be deleted.
-- range	Specifies the range of the object to be downloaded. This option value must be in the 3-9, 3-, or -9 format.
-- encoding - type	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
-- include	Includes objects that match a specified string, such as *.jpg.
-- exclude	Excludes objects that match a specified string, such as *.txt.
-- meta	Configures the metadata of an object in the [header:value#header:value...] format. Example: Cache-Control:no-cache#Content-Encoding:gzip For more information, see #unique_52 .
-- acl	Configures the ACL for an object. Default value: default. Valid values: <ul style="list-style-type: none"> · default: The object inherits the ACL for the bucket it belongs to. · private · public-read · public-read-write

Option	Description
<pre>-- snapshot - path</pre>	<p>If you specify the <code>-- snapshot - path</code> option when uploading or downloading multiple objects, ossutil takes a snapshot of the upload or download and stores the snapshot information in the specified directory. When this option is specified the next time objects are uploaded or downloaded, ossutil will read the snapshot information from the specified directory to perform incremental upload or download.</p> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p> Notice:</p> <ul style="list-style-type: none"> • The <code>-- snapshot - path</code> option is used in certain scenarios to accelerate the incremental upload or download of objects. This option can be used when the number of objects is large and no other users modify the corresponding objects in OSS during two uploads. This option cannot be used to copy objects. • The <code>-- snapshot - path</code> option records the local <code>lastModifiedTime</code> of uploaded or downloaded objects, and then compares that recorded <code>lastModifiedTime</code> with that of objects to be uploaded or downloaded next time to determine which objects can be skipped. When using this option, make sure that the corresponding objects in OSS are not modified during the two uploads or downloads. In other scenarios where objects are updated in OSS during the two uploads or downloads, use the <code>-- update</code> option to perform incremental upload or download on objects. • ossutil does not automatically delete snapshot information from the directory specified by <code>snapshot -path</code>. If the snapshot information is not necessary to retain, remove the directory. • Additional overheads are required to read and write snapshot information. We recommend that you do not use this option in any of the following scenarios: The number of objects to be uploaded or downloaded is small. The network is properly connected. Other users need to perform operations on those objects. In this case, you can use the <code>-- update</code> option to perform incremental upload or download. • The <code>-- update</code> and <code>-- snapshot - path</code> options can be used together. ossutil first uses the <code>--snapshot-path</code> option to determine whether to skip the upload or download of an object. If the upload or download is not skipped by <code>--snapshot-path</code>, ossutil then uses the <code>-- update</code> option to determine whether to skip the upload or download of the object. </div>
Issue: 20190920	55

Option	Description
-- disable - crc64	Disables CRC64. ossutil enables CRC64 during data transmission by default.
-- maxupspeed	Specifies the maximum upload speed in KByte/s. The default value is 0, which indicates that there is no maximum upload speed.
-- payer	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.
-- partition - download	Used to download object partitions. The value of this option is in the "partition number:total number of partitions" format. A value of 1:5 indicates that ossutil downloads partition 1 out of the total five partitions. Partitioning rules for objects are determined by ossutil. This option divides an object to be downloaded into multiple partitions that can be downloaded by multiple ossutil commands. Each ossutil command downloads its own partition. You can run multiple ossutil commands on different ECS instances in parallel.
- j , -- jobs	Specifies the number of concurrent operations performed across multiple objects. Valid values: 1 to 10000. Default value: 3.
-- parallel	Specifies the number of concurrent operations performed on a single object. Valid values: 1 to 10000. The value of this option is determined by ossutil based on the operation type and object size by default.
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
-- retry - times	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is http://120.79.**.**:3128 or socks5://120.79.**.**:1080.
-- proxy - user	Specifies the username of the proxy server. The default value is null.
-- proxy - pwd	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.10 create-symlink

The `create - symlink` command is used to create a symbolic link (also known as soft link).

**Note:**

For more information about symbolic links, see [#unique_77](#).

Command syntax

```
./ ossutil create - symlink cloud_url target_obj ect [-- encoding - type url ] [-- payer requester ] [- c file ]
```

Examples

```
./ ossutil create - symlink oss :// bucket1 / b oss :// bucket1 / path / a . txt
```

This command creates a symbolic link object named `b` for the `a . txt` object in the `path` folder of `bucket1`.

**Note:**

The system does not perform the following checks when symbolic links are created:

- Whether the mapped object exists.
- Whether the mapped object is valid.
- Whether the user has permission to access the mapped object.
- The preceding checks are performed when you attempt to call operations such as `GetObject` to access the mapped object.
- If the object to be added already exists and you have permissions to access the object, the newly added object will overwrite the existing object.

You can run the [#unique_54](#) or [#unique_46](#) command to view the object a symbolic link maps to.

Common options

The following table describes the options you can add to the `create - symlink` command.

Option	Description
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- retry - times</code>	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- payer</code>	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.**.**:3128</code> or <code>socks5://120.79.**. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.11 du

The `du` command is used to obtain the bucket capacity of a bucket or the size of a specified object or folder.

Command syntax

```
./ ossutil du oss :// bucket [/ prefix ] [-- payer requester ]
```

Examples

- Query the capacity of a specify bucket

```
./ ossutil du oss :// bucket1
object count : 9    object sum size : 471075
part count : 0     part sum size : 0
```

```
total du size ( byte ): 471075
```

- View the size of a specified folder

```
./ ossutil du oss :// bucket1 / test /
object count : 2 object sum size : 64482
part count : 0 part sum size : 0
total du size ( byte ): 64482

0 . 176018 ( s ) elapsed
```

- View the size of a specified object

```
./ ossutil du oss :// bucket1 / test . txt
object count : 1 object sum size : 27856
part count : 0 part sum size : 0
total du size ( byte ): 27856

0 . 241024 ( s ) elapsed
```

- View the size of all objects whose names start with a specified prefix

```
./ ossutil du oss :// bucket1 / test
object count : 3 object sum size : 92338
part count : 0 part sum size : 0
total du size ( byte ): 92338

0 . 184018 ( s ) elapsed
```

Common options

The following table describes the options you can add to the `du` command.

Option	Description
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> • info: generates prompt logs. • debug: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.**.**:3128</code> or <code>socks5://120.79.**.**:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.
<code>-- payer</code>	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.12 getallpartsize

The `getallpart size` command is used to obtain the size of each part generated in incomplete multipart upload tasks initiated to objects in a bucket, and the total size of all parts.

Command syntax

```
./ ossutil getallpart size oss :// bucket
```

Examples

List the size of each part generated in incomplete multipart upload tasks initiated to objects in a bucket and the total size of all parts.

```
./ ossutil getallpart size oss :// bucket1
PartNumber UploadId Size (
Byte ) Path
1 F18A92392D FD4B3FA897 C267829FE4 17
52428800 oss :// bucket1 / test / test1 . txt
2 F18A92392D FD4B3FA897 C267829FE4 17
52428800 oss :// bucket1 / test / test1 . txt
3 F18A92392D FD4B3FA897 C267829FE4 17
52428800 oss :// bucket1 / test / test1 . txt
4 F18A92392D FD4B3FA897 C267829FE4 17
52428800 oss :// bucket1 / test / test1 . txt
5 F18A92392D FD4B3FA897 C267829FE4 17
52428800 oss :// bucket1 / test / test1 . txt
6 F18A92392D FD4B3FA897 C267829FE4 17
52428800 oss :// bucket1 / test / test1 . txt

total part count : 6 total part size ( MB ): 300 . 00
```

Common options

The following table describes the options you can add to the `getallpart size` command.

Option	Description
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code> . If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79. **: **:1080</code> .

Option	Description
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.13 hash

The `hash` command is used to calculate the CRC64 or MD5 value of a local file.

Command syntax

```
./ ossutil hash file_url [-- type = hashtype ]
```

`-- type` specifies the algorithm that is used for calculation. Valid values: `crc64` and `md5`. Default value: `crc64`.



Note:

- You can run the [#unique_54](#) command to view the CRC64 or Content-MD5 value of objects in OSS. The `X - Oss - Hash - Crc64ecma` field indicates the CRC64 value. The `Content - Md5` field indicates the MD5 value.
- If the object was uploaded before OSS supports CRC64, the `stat` command may fail to obtain the CRC64 value.
- The `stat` command may fail to obtain the Content-MD5 value of objects that are uploaded through append upload or multipart upload.
- If `--type` is set to `md5`, the MD5 and Content-MD5 values 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 encoding the number in Base64.
- The CRC64 value is calculated based on [Standard ECMA-182](#).
- For more information about Content-MD5, see [RFC 1864](#).

Examples

- Calculate the CRC64 value of a local file

```
./ ossutil hash test . txt -- type = crc64
```

```
CRC64 - ECMA : 2959929367 43767023
```

- Calculate the MD5 value of a local file

```
./ ossutil hash test . txt -- type = md5
MD5 : 01C3C45C03 B2AF225EFA
D9F911A33D 73
Content - MD5 : AcPEXA0yry Je + tn5EaM9cw ==
```

Common options

The following table describes the options you can add to the `hash` command.

Option	Description
<code>-- type</code>	Specifies the algorithm that is used for calculation. Valid values: <code>crc64</code> and <code>md5</code> . Default value: <code>crc64</code> .
<code>-- loglevel</code>	Specifies the log level. The default value is <code>null</code> , indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> • <code>info</code>: generates prompt logs. • <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.



Note:

For more information about common options, see [#unique_62](#).

3.3.14 help

The `help` command is used to obtain the help information about commands. We recommend that you run the `help` command when you want to know the usage of a command.

Command syntax

```
./ ossutil help [ command ]
```

Examples

- Obtain the help information about all commands

```
./ ossutil help
```

- Obtain the help information about the `cp` command

```
./ ossutil help cp
```

- View the information about all options

```
./ ossutil help -h
```

- Display the help information about the `ls` command in Chinese

```
./ ossutil help ls -L ch
```

3.3.15 lifecycle

The `lifecycle` command is used to add, modify, query, or delete lifecycle configurations.



Note:

For more information about lifecycle rules, see [#unique_83](#).

Command syntax

- Add or modify lifecycle configurations

```
./ ossutil lifecycle -- method put oss :// bucket  
local_xml_ file [ options ]
```

`ossutil` reads the configurations in `local_xml_ file`, and adds the rules obtained from the configuration file for the bucket. If the rule ID exists in the configuration file, `ossutil` will overwrite the configurations of the existing rule ID.



Note:

A lifecycle configuration file is an XML file that describes the rules on handling expired objects, such as whether to delete the object or to change the storage class of expired objects. You can configure the rules based on your needs. The following code provides an example of a complete lifecycle configuration:

```
<? xml version = " 1 . 0 " encoding = " UTF - 8 " ? >
< LifecycleC onfigurati on >
  < Rule >
    < ID > RuleID </ ID >
    < Prefix > Prefix </ Prefix >
    < Status > Status </ Status >
    < Expiration >
      < Days > Days </ Days >
    </ Expiration >
    < Transition >
      < Days > Days </ Days >
      < StorageCla ss > StorageCla ss </ StorageCla ss >
    </ Transition >
    < AbortMulti partUpload >
      < Days > Days </ Days >
    </ AbortMulti partUpload >
  </ Rule >
</ LifecycleC onfigurati on >
```

For more information about lifecycle configurations, see [#unique_84](#).

- Obtain lifecycle configurations

```
ossutil lifecycle -- method get oss :// bucket [ local_file ]
```

The `local_file` parameter specifies the name of the configuration file. If this parameter is specified, ossutil saves the obtained lifecycle configurations as a local file. If this parameter is not specified, ossutil displays the obtained lifecycle configurations.

- Delete lifecycle configurations

```
./ ossutil lifzecycle -- method delete oss :// bucket
```

Examples

- Add a lifecycle rule that changes the storage class of objects uploaded for 10 days and prefixed by `test /` to IA

```
./ ossutil lifecycle -- method put oss :// bucket1 / file /
lifecycle . xml
```

The content of the `lifecycle . xml` configuration file is as follows:

```
<? xml version = " 1 . 0 " encoding = " UTF - 8 " ? >
< LifecycleC onfigurati on >
  < Rule >
```

```
< ID > 1 </ ID >
< Prefix > test </ Prefix >
< Status > Enabled </ Status >
< Transition >
  < Days > 10 </ Days >
  < StorageClass > IA </ StorageClass >
</ Transition >
</ Rule >
</ LifecycleConfiguration >
```

- Add a lifecycle rule that deletes objects modified before January 1, 2019 and prefixed by `test /`

```
./ossutil lifecycle --method put oss://bucket1 /file /
lifecycle.xml
```

The content of the `lifecycle.xml` configuration file is as follows:

```
<?xml version="1.0" encoding="UTF-8"? >
< LifecycleConfiguration >
  < Rule >
    < ID > 2 </ ID >
    < Prefix > test </ Prefix >
    < Status > Enabled </ Status >
    < Expiration >
      < CreatedBeforeDate > 2014-10-11T00:00:00.000Z </
CreatedBeforeDate >
    </ Expiration >
  </ Rule >
</ LifecycleConfiguration >
```

- Obtain lifecycle configurations

```
./ossutil lifecycle --method get oss://bucket1 /file /
lifecycle.xml
```

- Delete lifecycle configurations

```
./ossutil lifecycle --method delete oss://bucket1
```

Common options

The following table describes the options you can add to the `lifecycle` command.

Option	Description
<code>--method</code>	<p>Specifies the HTTP request method. Valid values:</p> <ul style="list-style-type: none"> • <code>put</code>: adds or modifies configurations • <code>get</code>: obtains configurations • <code>delete</code>: deletes configurations

Option	Description
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · <code>info</code>: generates prompt logs. · <code>debug</code>: generates detailed logs that contain HTTP requests and responses.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 proxies are supported. An example of the URL is <code>http://120.79. **. **:3128</code> or <code>socks5://120.79. **. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.16 listpart

The `listpart` command is used to list information of parts that are generated when multiplart upload tasks fail.

Command syntax

```
./ ossutil listpart oss :// bucket / object uploadid [ options ]
```

Before you run this command to obtain information of parts, run the `ls -m` command to obtain information of upload IDs corresponding to buckets and objects.

Examples

Lists parts in an object that are generated when multipart upload tasks have not been completed.

```
./ ossutil listpart oss :// bucket / object 15754AF798
0C4DFB8193 F190837520 BB
```

Common options

The following table describes the options that you can add to the `listpart` command.

Option	Description
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79. **. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.17 logging

The `logging` command is used to add, modify, query, or delete the logging configurations for a specified bucket.



Note:

For more information about logging, see [#unique_87](#).

Command syntax

- Add or modify logging configurations

```
./ ossutil logging -- method put oss :// bucket oss ://
target - bucket /[ prefix ]
```

If logging is disabled for the bucket, you can run this command to save access logs as objects in the bucket specified by the `target-bucket` parameter. However, if logging is enabled for the bucket, you can run this command to change the directory in which access logs are stored.

The `prefix` parameter specifies the prefix of access logs and the directory where bucket access logs are stored. If this parameter is specified, `ossutil` saves access logs to the directory of the bucket specified by the `target-bucket` parameter. If this parameter is not specified, `ossutil` saves your access logs to the root directory of the

bucket specified by the `target-bucket` parameter. For more information about log file naming conventions, see [Set logging](#).

- Obtain logging configurations

```
./ ossutil logging -- method get oss :// bucket [
local_xml_ file ]
```

The `local_xml_ file` parameter specifies the name of the configuration file. If this parameter is specified, ossutil saves the obtained logging configurations as a local file. If this parameter is not specified, ossutil displays the obtained logging configurations.

- Delete logging configurations

```
./ ossutil logging -- method delete oss :// bucket
```

Examples

- Add logging configurations

```
./ ossutil logging -- method put oss :// bucket1 oss ://
bucket2 / logging
```

- Obtain logging configurations

```
./ ossutil logging -- method get oss :// bucket1 / file /
logging . xml
```

- Delete logging configurations

```
./ ossutil logging -- method delete oss :// bucket1
```

Common options

The following table describes the options you can add to the `logging` command.

Option	Description
<code>-- method</code>	<p>Specifies the HTTP request method. Valid values:</p> <ul style="list-style-type: none"> • <code>put</code>: adds or modifies logging configurations. • <code>get</code>: obtains logging configurations. • <code>delete</code>: deletes logging configurations.
<code>-- loglevel</code>	<p>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</p> <ul style="list-style-type: none"> • <code>info</code>: generates prompt logs. • <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.

Option	Description
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is http://120.79.***:3128 or socks5://120.79. **: **:1080.
-- proxy - user	Specifies the username of the proxy server. The default value is null.
-- proxy - pwd	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.18 ls

The `ls` command is used to list buckets, objects, or parts.

Command syntax

```
./ ossutil ls [ oss :// bucket [/ prefix ]] [- s ] [- d ] [--
limited - num num ] [-- marker marker ] [-- upload - id - marker
umarker ] [-- payer requester ] [- c file ] [-- include
include - pattern ] [-- exclude exclude - pattern ] [-- payer
requester ]
```

Examples

- List buckets

```
./ ossutil ls
CreationTi me                               Region
StorageCla ss      BucketName
2016 - 10 - 2116 : 18 : 37 + 0800 CST      oss - cn -
hangzhou           Archive      oss :// bucket1
2016 - 12 - 0115 : 06 : 21 + 0800 CST      oss - cn -
hangzhou           Standard    oss :// bucket2
2016 - 07 - 2010 : 36 : 24 + 0800 CST      oss - cn -
hangzhou           IA         oss :// bucket3
Bucket Number is : 3
0 . 252174 ( s ) elapsed
```



Note:

You can also use the `./ ossutil ls oss ://` command to list buckets.

- List buckets by page

```
./ ossutil ls oss :// -- limited - num = ${ num } -- marker = ${
bucketname }
```

When the number of buckets is large, you can use the `-- limited - num` and `-- marker` options to list buckets by page.

- `-- limited - num` is used to control the number of entries displayed on each page.
- `-- marker` specifies the bucket that you want to start listing from. The results displayed by ossutil are returned starting from the first bucket in alphabetical order after the marker value is set. In most cases, this parameter value is set to the last bucket name displayed on the previous page.

The following code provides an example of how to list the first two buckets by page:

```
./ ossutil ls oss :// -- limited - num = 1 - s
oss :// bucket1
Bucket Number is : 1
0 . 303869 ( s ) elapsed
./ ossutil ls oss :// -- limited - num = 1 - s -- marker =
bucket1
oss :// bucket2
Bucket Number is : 1
0 . 257636 ( s ) elapsed
```

- List all objects in a specified bucket

```
./ ossutil ls oss :// bucket1
LastModifi edTime          Size ( B )  StorageCla ss
ETAG          ObjectName
2016 - 12 - 0115 : 06 : 37 + 0800  CST          10363812
Standard      61DE142E5A  FF9A674870  7D4A77BFBC  FB          oss
:// bucket1 / a1
2016 - 12 - 0115 : 06 : 42 + 0800  CST          10363812
Standard      61DE142E5A  FF9A674870  7D4A77BFBC  FB          oss
:// bucket1 / a2
2016 - 12 - 0115 : 06 : 45 + 0800  CST          10363812
Standard      61DE142E5A  FF9A674870  7D4A77BFBC  FB          oss
:// bucket1 / a3
Object Number is : 3
```

```
0 . 007379 ( s ) elapsed
```

- List all objects by page

```
./ ossutil ls oss :// bucket -- limited - num = ${ num } --
marker = ${ obj }
```

Similar to the operation of listing buckets by page, you can use the `-- limited - num` and `-- marker` options to list objects by page. The following code provides an example of how to list the first two objects in the `bucket1` root directory:

```
./ ossutil ls oss :// bucket1 -- limited - num = 1
LastModifi edTime          Size ( B )  StorageCla ss
      ETAG                ObjectName
2016 - 12 - 0115 : 06 : 37 + 0800  CST        10363812
Standard      61DE142E5A FF9A674870 7D4A77BFBC FB          oss
:// bucket1 / a1
Object Number is : 1
0 . 007379 ( s ) elapsed
$./ ossutil ls oss :// bucket1 -- limited - num = 1 -- marker
= a1
LastModifi edTime          Size ( B )  StorageCla ss
      ETAG                ObjectName
2016 - 12 - 0115 : 06 : 42 + 0800  CST        10363812
Standard      61DE142E5A FF9A674870 7D4A77BFBC FB          oss
:// bucket1 / a2
Object Number is : 1
0 . 008392 ( s ) elapsed
```

- List objects that meet specified conditions

You can set `-- include /-- exclude` to list objects that meet specified conditions. For more information about `-- include /-- exclude`, see [cp](#).

- List all objects whose format is not `jpg`

```
./ ossutil ls oss :// my - bucket / path -- exclude "*. jpg"
" - r
```

- List all objects that contain `abc` in their names and are not in the `jpg` or `txt` format

```
./ ossutil ls oss :// my - bucket1 / path -- include "* abc"
* " -- exclude "*. jpg " -- exclude "*. txt " - r
```

- List upload IDs

Add the `- m` option to the `ls` command to list upload IDs of all incomplete multipart upload tasks initiated to objects with a specified prefix.

```
./ ossutil ls oss :// bucket1 / obj1 - m
InitiatedT ime          UploadID
      ObjectName
```

```

2017 - 01 - 13  03 : 45 : 26  + 0000  CST      15754AF798
0C4DFB8193 F190837520 BB      oss :// bucket1 / obj1
2017 - 01 - 13  03 : 43 : 13  + 0000  CST      2A1F9B4A95
E341BD9285 CC42BB950E E0      oss :// bucket1 / obj1
UploadId      Number      is : 2
0 . 070070 ( s ) elapsed

```

- List all objects and incomplete multipart upload tasks

You can add the `-a` option to the `ls` command to list the upload IDs of all incomplete multipart upload tasks initiated to objects with a specified prefix and the uploaded objects with the prefix.

```

./ ossutil  ls  oss :// bucket1  - a
LastModifi edTime                               Size ( B )   StorageCla ss
ETAG                                             ObjectName
2015 - 06 - 0514 : 06 : 29  + 0000  CST          201933
Standard      7E2F4A7F1A C9D2F0996E  8332D5EA5B  41           oss
:// bucket1 / dir1 / obj1
2015 - 06 - 0514 : 36 : 21  + 0000  CST          201933
Standard      6185CA2E8E B8510A61B3  A845EAFE41  74           oss
:// bucket1 / obj1
2016 - 04 - 0814 : 50 : 47  + 0000  CST          6476984
Standard      4F16FDAE7A C404CEC8B7  27FCC67779  D6           oss
:// bucket1 / sample . txt
Object      Number      is : 3
InitiatedT ime                               UploadID
ObjectName
2017 - 01 - 1303 : 45 : 26  + 0000  CST          15754AF798
0C4DFB8193 F190837520 BB      oss :// bucket1 / obj1
2017 - 01 - 1303 : 43 : 13  + 0000  CST          2A1F9B4A95
E341BD9285 CC42BB950E E0      oss :// bucket1 / obj1
2017 - 01 - 1303 : 45 : 25  + 0000  CST          3998971ACA
F94AD9AC48 EAC1988BE8 63      oss :// bucket1 / obj2
2017 - 01 - 2011 : 16 : 21  + 0800  CST          A20157A7B2
FEC4670626 DAE0F4C007 3C      oss :// bucket1 / tobj
UploadId      Number      is : 4
0 . 191289 ( s ) elapsed

```

- Display the listed results in short format

```

./ ossutil  ls  oss :// bucket1  - s
oss :// bucket1 / a1
oss :// bucket1 / a2
oss :// bucket1 / a3
Object      Number      is : 3
0 . 007379 ( s ) elapsed

```

- List buckets in the form of directories

Use the `-d` option to list objects and subdirectories in the current directory, rather than recursively displaying all objects in all subdirectories. Example:

```

./ ossutil  ls  oss :// bucket1  - s  - d
oss :// bucket1 / obj1
oss :// bucket1 / sample . txt
oss :// bucket1 / dir1 /
Object and Directory      Number      is : 3

```

```
0 . 119884 ( s ) elapsed
```

- List objects or subdirectories with a specified prefix in the current directory

```
./ ossutil ls oss :// bucket1 / test - d
oss :// bucket1 / test . jpg
oss :// bucket1 / test /
Object and Directory Number is : 2
```

- List buckets in the pay-by-requester mode

```
./ ossutil ls oss :// bucket -- payer = requester
```

Common options

The following table describes the options you can add to the `ls` command to specify the items to be listed and how they can be listed.

Option	Description
- s	Lists items in short format.
- d	Lists objects and subdirectories in the current directory, rather than recursively displaying all objects in all subdirectories.
- m	Lists only incomplete multipart upload tasks in the bucket. Objects are not displayed.
- a	Lists both objects and incomplete multipart upload tasks in the bucket.
-- limited - num	Specifies the maximum number of returned results. This option can be used together with the <code>-- marker</code> option to list objects by page.
-- marker	Specifies the bucket name, object name, or multipart upload ID that you want to start listing from.
-- upload - id - marker	Specifies the multipart upload ID that you want to start listing from.
-- payer	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.
-- encoding - type	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.

Option	Description
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
-- include	Includes objects that match a specified string, such as *.jpg.
-- exclude	Excludes objects that match a specified string, such as *.txt.
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is http://120.79.**.**:3128 or socks5://120.79.**.**:1080.
-- proxy - user	Specifies the username of the proxy server. The default value is null.
-- proxy - pwd	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.19 mb

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

Command syntax

```
./ ossutil mb oss :// bucketname [-- acl = ACL ][-- storage -
class sc ][- c file ]
```

Examples

- Create a bucket

```
./ ossutil mb oss :// bucket1
```

**Notice:**

The bucket name must be unique. If a bucket of the specified name already exists, an error message will be displayed.

- Specify the access control list (ACL) when creating a bucket

```
./ ossutil mb oss :// bucket1 -- acl = public - read - write
```

- Specify the storage class when creating a bucket

```
./ ossutil mb oss :// bucket1 -- storage - class IA
```

- Create a bucket in a specified region

```
./ ossutil mb oss :// bucket1 - e oss - cn - beijing .
aliyuncs . com
```

- Create a bucket from a specified configuration file

```
./ ossutil mb oss :// bucket1 - c your_confi g_file_pat h
```

Common options

The following table describes the options you can add to the `mb` command to specify bucket attributes.

Option	Description
<code>-- acl</code>	Specifies the ACL for the bucket. Default value: <code>private</code> . Valid values: <ul style="list-style-type: none"> • <code>private</code> • <code>public-read</code> • <code>public-read-write</code> For more information about ACLs, see #unique_15 .
<code>-- storage - class</code>	Specifies the default storage class of the bucket. Default value: <code>Standard</code> . Valid values: <ul style="list-style-type: none"> • <code>Standard</code> • <code>IA</code> • <code>Archive</code> For more information about storage classes, see #unique_16 .
<code>- c , -- config - file</code>	Specifies the configuration file used to create the bucket. If this option is not specified, the default configuration file is used to create the bucket. For more information about the configuration file, see config .
<code>- e , -- endpoint</code>	Specifies the endpoint corresponding to the region where the bucket resides. If this option is not specified, the endpoint in the default configuration file is used to specify the region of the bucket.

Option	Description
- L , -- language	Specifies the language ossutil uses. Valid values: CH and EN. Default value: CH. If you plan to set this option to CH, ensure that your system supports UTF-8 encoding.
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
-- retry - times	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is http://120.79.***:3128 or socks5://120.79. **. **:1080.
-- proxy - user	Specifies the username of the proxy server. The default value is null.
-- proxy - pwd	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.20 mkdir

The `mkdir` command is used to create a directory in a bucket.

Command syntax

```
./ ossutil mkdir oss :// bucket / dir /
```

Examples

- Create a directory

```
./ ossutil mkdir oss :// bucket1 / dir1 /
```

**Note:**

- A directory must end with a forward slash (/). If the specified directory name does not end with a forward slash, ossutil will automatically add one forward slash (/).

- If the specified directory already exists, an error is displayed.

- Create a multi-level directory

```
./ ossutil mkdir oss :// bucket / dir1 / dir2 /
```

When creating a multi-level directory, ossutil only creates the last level of directory. If `dir2 /` is deleted and there are no objects in `dir1 /`, `dir1 /` will also be deleted after the command is run.

Common options

The following table describes the options you can add to the `mkdir` command.

Option	Description
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code> . If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- loglevel</code>	Specifies the log level. The default value is <code>null</code> , indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> • <code>info</code>: generates prompt logs. • <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79. **: **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is <code>null</code> .
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is <code>null</code> .



Note:

For more information about common options, see [#unique_62](#).

3.3.21 object-tagging

The `object - tagging` command is used to add, modify, query, or delete tagging configurations for objects.



Note:

For more information about object tagging, see [#unique_93](#).

Command syntax

- Add or modify object tagging configurations

```
./ ossutil object - tagging -- method put oss :// bucket [/
prefix ] key # value [-- encoding - type url ] [- r ] [-- payer
requester ] [-- version - id versionId ] [- c file ]
```

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



Note:

- A maximum of 10 tags can be set for each object. Each object tag must have a unique tag key.
- A tag key can be maximum of 128 characters in length. A tag value can be a maximum of 256 characters in length.
- Keys and values are case-sensitive.
- The key and value of the tag can contain letters, digits, spaces, and special characters such as
+ = . _ : /
- Only the bucket owner and authorized users have the read and write permissions on object tags. These permissions are independent of object ACLs.
- During cross-region replication, object tags are also replicated to the destination bucket.

- Query object tagging configurations

```
./ ossutil object - tagging -- method get oss :// bucket [/
prefix ] [-- encoding - type url ] [- r ] [-- payer requester
] [-- version - id versionId ] [- c file ]
```

- Delete object tagging configurations

```
./ ossutil object - tagging -- method delete oss ://
bucket [/ prefix ] [-- encoding - type url ] [- r ] [-- payer
requester ] [-- version - id versionId ] [- c file ]
```

Examples

- Add object tagging configurations

```
./ ossutil object - tagging -- method put oss :// bucket1 /
test . jpg a # 1 b # 2 c # 3
```

```
0 . 168034 ( s ) elapsed
```

- Query object tagging configurations

```
./ ossutil object - tagging -- method get oss :// bucket1 /
test . jpg
object index tag index tag key tag value
object
-----
1 0 " a " " 1 " 123 / test
. jpg
1 1 " b " " 2 " 123 / test
. jpg
1 2 " c " " 3 " 123 / test
. jpg
0 . 228023 ( s ) elapsed
```

- Delete object tagging configurations

```
./ ossutil object - tagging -- method delete oss :// bucket1
/ test . jpg
0 . 200020 ( s ) elapsed
```

Common options

The following table describes the options you can add to the `object - tagging` command.

Option	Description
<code>-- method</code>	Specifies the HTTP request method. Valid values: <ul style="list-style-type: none"> • <code>put</code>: adds or modifies object tagging configurations. • <code>get</code>: obtains object tagging configurations. • <code>delete</code>: deletes object tagging configurations.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> • <code>info</code>: generates prompt logs. • <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.**.**:3128</code> or <code>socks5://120.79.**.**:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.

Option	Description
-- payer	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.
- j , -- jobs	Specifies the number of concurrent tasks when multiple objects are operated. Valid values: 1 to 10000. Default value: 3.
- r , -- recursive	Recursively performs operations on objects in a bucket. If this option is specified, commands that support this option will perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, the commands will only perform operations on a single specified object.
-- encoding - type	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.22 probe

The `probe` command is used to monitor access to OSS, and troubleshoot problems caused during the upload and download process by network faults or incorrect parameter configurations.

Command syntax

- Download an object from a bucket to your local device using the object URL and generate a troubleshooting report

```
./ossutil probe --download --url http_url [--addr =
domain_name] [file_name]
```

After downloading an object from a bucket to your local device using the object URL, you can test the network transmission quality and generate a troubleshooting report.

- `--url` : specifies the URL of an object in a specified bucket.

- If the ACL for the object is `public-read`, the URL does not carry a signature.

Example: `https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg`.

- If the ACL for the object is `private`, the URL carries a signature and starts and ends with a double quotation mark ("). **Example:** “`https://bucketname`

`.oss-cn-beijing.aliyuncs.com/myphoto.jpg?Expires=1552015472&OSSAccessKeyId=TMPxxxxxxxx5r9f1FV12y8_Qis6LUVmvoSCUSS7aboCCHtydQ0axN32Sn-UvyY3AAAwLAIUarYNLc087AKMEcE503AxxxxxxxxoCFAQuRdZYyVFyq0W8QkGAN-bamUiQ&Signature=bIa4llbMbl drl7rwckr%20FXxvTtxw%3D”`.



Note:

For more information about how to obtain the URL of an object, see [Obtain URLs](#).

- `-- addr = domain_name` : optional. This parameter specifies the domain name or IP address to ping while the object is being downloaded. If this parameter is not specified, no ping operations are performed.
 - If the `-- addr = option` is set to the default value, ossutil pings `www.aliyun.com`.
 - If the `-- addr = option` is set to a domain name or IP address, ossutil pings the specified domain name or IP address.
- `file_name` : optional. This parameter specifies the file path in which to store the downloaded object. If `file_name` is not specified, ossutil saves the downloaded object to the current directory and determines the object name. If you use `file_name` to specify an object or directory name, ossutil uses the specified `file_name` value to name the downloaded object or saves the downloaded object to the directory specified by `file_name`.

- Download an object from a bucket and generate a troubleshooting report

```
./ossutil probe --download --bucketname bucket-name [--object=object_name] [--addr=domain_name] [file_name]
```

- `-- bucketname` : specifies the name of the bucket that contains the object to be downloaded.
- `-- object =`: optional. This parameter specifies the file path where the downloaded object is stored. Example: `path / myphoto . jpg`. If `-- object =` is not specified, ossutil generates a temporary object, uploads the object to the bucket specified by the `bucket-name` parameter, and then downloads the object. After this object is downloaded, ossutil deletes this object from your local device and bucket.

- Upload an object and generate a troubleshooting report

```
./ossutil probe --upload [file_name] --bucketname bucket-name [--object=object_name] [--addr=domain_name] [--upmode]
```

- `file_name` : optional. This parameter specifies the name of the object that you want to upload to the bucket specified by the `bucket-name` parameter. If `file_name` is not specified, ossutil generates a temporary object and uploads it to

the specified bucket. After completing the probe, ossutil deletes this temporary object.

- `-- object =:` optional. This parameter specifies the name of an object or directory. Example: `path / myphoto . jpg` . If `-- object =` is not specified, ossutil generates a name for the uploaded object. After completing the probe, ossutil deletes this object.
- `-- upmode :` optional. This parameter specifies the upload method. Default value: normal. Valid values:
 - normal
 - append
 - multipart

Examples

- Download an object from a bucket to your local device using the object URL and generate a troubleshooting report

```
./ ossutil probe -- download -- url https :// bucket1 . oss
- cn - beijing . aliyuncs . com / myphoto . jpg -- addr = www .
aliyun . com / file / myphoto . jpg
```

- Download an object from a bucket and generate a troubleshooting report

```
./ ossutil probe -- download -- bucketname bucket1 -- object
= myphoto . jpg -- addr = www . aliyun . com & nbsp ;/ file /
myphoto . jpg
```

- Check the upload result and generate a troubleshooting report

```
./ ossutil probe -- upload / file / myphoto . jpg -- bucketname
bucket1 -- object = myphoto . jpg & nbsp ;-- upmode normal
```

View a troubleshooting report

After running the `probe` command, you can view each task execution procedure and overall upload or download results.

- If an `×` is displayed following a procedure, the operation failed.
- If no `×` is displayed following a procedure, the operation succeeded. If the upload or download was successful, ossutil displays the object size and the time the object

was uploaded or downloaded. If the upload or download failed, ossutil displays the failure cause or troubleshooting advice.



Note:

ossutil may not generate troubleshooting advice for a few errors. In this case, you can troubleshoot the problems based on the error codes by following the instructions provided in [OSS error codes](#).

After running the `probe` command, ossutil generates an object whose name begins with `logOssProbe` in your current directory. This object contains details about the commands that you have run to troubleshoot problems.

Common options

The following table describes the options you can add to the `probe` command.

Option	Description
<code>-- url</code>	Specifies the network address of the object. ossutil will use this address to download the object.
<code>-- bucketname</code>	Specifies the name of the bucket in OSS.
<code>-- object</code>	Specifies the name of the object in OSS.
<code>-- addr</code>	Specifies the domain name to ping. Default value: <code>www.aliyun.com</code> .
<code>-- upmode</code>	Specifies the upload method. Default value: <code>normal</code> . Valid values: <ul style="list-style-type: none"> · <code>normal</code> · <code>append</code> · <code>multipart</code>
<code>-- upload</code>	Uploads the object to OSS.
<code>-- download</code>	Downloads the object from OSS.
<code>-- loglevel</code>	Specifies the log level. The default value is <code>null</code> , indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · <code>info</code>: generates prompt logs. · <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.

Option	Description
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79. **: **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.23 read-symlink

The `read - symlink` command is used to read the description of a symbolic link object.



Note:

- For more information about symbolic links, see [#unique_77](#).
- For more information about the command used to create symbolic links, see [#unique_33](#).

Command syntax

```
./ ossutil read - symlink oss :// bucket / object [-- encoding -
type url ] [-- payer requester ] [- c file ]
```

This operation requires that you have the read permission on the symbolic link object. In the output, X-Oss-Symlink-Target indicates the object the symbolic link points to. If the object to be managed is not a symbolic link object, NotSymlink is returned.

Examples

Obtain the symbolic link information of object1.

```
./ ossutil read - symlink oss :// bucket1 / object1
Etag : 455E20DBFF F1D588B67D 092C46B16D B6
Last - Modified &nbsp; ; &nbsp; ; &nbsp; ; &nbsp; ; &nbsp; ; : 2017
- 04 - 17 14 : 49 : 42 + 0800 CST
```

```
X - Oss - Symlink - Target &nbsp; ; &nbsp; ;: a
```

Common options

The following table describes the options you can add to the `create - symlink` command.

Option	Description
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code> . If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- loglevel</code>	Specifies the log level. The default value is <code>null</code> , indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> <code>info</code>: generates prompt logs. <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- retry - times</code>	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
<code>-- payer</code>	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to <code>requester</code> .
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.**.**:3128</code> or <code>socks5://120.79.**. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is <code>null</code> .
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is <code>null</code> .



Note:

For more information about common options, see [#unique_62](#).

3.3.24 referer

The `referer` command is used to add, modify, query, or delete hotlink protection configurations for a bucket.



Note:

For more information about hotlink protection, see [#unique_98](#).

Command syntax

- Add or modify hotlink protection configurations for a specified bucket

```
./ ossutil referer -- method put oss :// bucket referer -  
value [-- disable - empty - referer ]
```

If the specified bucket does not have hotlink protection configured, you can run this command to add hotlink protection configurations. If the specified bucket has hotlink protection configured, you can run this command to overwrite the existing hotlink protection configurations.

- `referer - value` : specifies a Referer whitelist. Only the specified domain names are allowed to access OSS resources. Asterisks (*) and question marks (?) are supported as wildcards. Multiple domain names must be separated with spaces.
 - `-- disable - empty - referer` : specifies whether the Referer field can be left empty in an access request. If this option is specified, the Referer field cannot be left empty. If the Referer field is required, you must include the Referer field in the HTTP or HTTPS request header to access OSS resources.
- Obtain the hotlink protection configurations of a bucket

```
./ ossutil referer -- method get oss :// bucket [  
local_xml_ file ]
```

The `local_xml_ file` parameter specifies the name of the configuration file. If this parameter is specified, the hotlink protection configurations are saved as a local file. If this parameter is not specified, ossutil displays the hotlink protection configurations.

- Delete the hotlink protection configurations of a bucket

```
./ ossutil referer -- method delete oss :// bucket
```

Examples

- Configure hotlink protection and specify the Referer field

```
./ ossutil referer -- method put oss :// bucket1 www .
test1 . com www . test2 . com -- disable - empty - referer
```

- Obtain the hotlink protection configurations of a bucket

```
./ ossutil referer -- method get oss :// bucket1 / file /
referer . xml
```

- Delete the hotlink protection configurations of a bucket

```
./ ossutil referer -- method delete oss :// bucket1
```

Common options

The following table describes the options you can add to the `referer` command.

Option	Description
<code>-- method</code>	Specifies the HTTP request method. Valid values: <ul style="list-style-type: none"> • <code>put</code>: adds or modifies hotlink protection configurations. • <code>get</code>: obtains hotlink protection configurations. • <code>delete</code>: deletes hotlink protection configurations.
<code>-- disable - empty - referer</code>	Specifies whether the Referer field can be left empty. If this option is specified, the Referer field cannot be left empty. If the Referer field is required, you must include the Referer field in the HTTP or HTTPS request header to access OSS resources.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> • <code>info</code>: generates prompt logs. • <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***.***:3128</code> or <code>socks5://120.79. **. **.1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.

Option	Description
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.25 restore

The `restore` command is used to restore an object from the frozen state to the readable state.

Command syntax

```
./ ossutil restore cloud_url [-- encoding - type url ] [-- payer requester ] [- r ] [- f ] [-- output - dir = odir ] [- c file ]
```

Examples

- Restore an object from the frozen state to the readable state

```
./ ossutil restore oss :// bucket / object
```

- Restore all objects that have a specified prefix to the readable state

```
./ ossutil restore oss :// bucket / path / - r
```

**Note:**

- The `restore` operation takes about one minute. You cannot read an object that is being restored.
- By default, the object will remain in the readable state for one day. Running the `restore` command for an object in the readable state will prolong the period for another day. You can prolong this period to a maximum of seven days. After this period ends, the object returns to the frozen state.

Common options

The following table describes the options you can add to the `restore` command.

Option	Description
- r , -- recursive	Recursively performs operations on objects in a bucket. If this option is specified, commands that support this option will perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, the commands will only perform operations on a single specified object.
- f , -- force	Forces an operation without prompting the user for confirmation.
- j , -- jobs	Specifies the number of concurrent tasks when multiple objects are operated. Valid values: 1 to 10000. Default value: 3.
-- encoding - type	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
-- output - dir =	Specifies the directory in which output objects are located. Output objects include report objects generated due to errors that occur when you use the <code>cp</code> command to copy multiple objects. For more information about the <code>report</code> objects, see the help information of the <code>cp</code> command. The default value is the <code>ossutil_output</code> directory in the current directory.
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
-- retry - times	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
-- payer	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79.***:1080</code> .
-- proxy - user	Specifies the username of the proxy server. The default value is null.
-- proxy - pwd	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.26 request-payment

The `request - payment` command is used to configure the pay-by-requester mode for buckets, or query the pay-by-requester configurations for buckets.

**Note:**

For more information about the pay-by-requester mode, see [#unique_101](#).

Command syntax

- Set the pay-by-requester mode

```
./ ossutil request - payment -- method put oss :// bucket
payment_pa ramer
```

Valid values of `payment_pa ramer` are `Requester` and `BucketOwne r`.

- `Requester` : enables the pay-by-requester mode. Requesters pay the cost of requests and the data download from buckets.
- `BucketOwne r` : disables the pay-by-requester mode. Bucket owners pay the cost of requests and the data download from their buckets.
- Query the pay-by-requester configurations

```
./ ossutil request - payment -- method get oss :// bucket
```

Examples

- Enable the pay-by-requester mode

```
./ ossutil request - payment -- method put oss :// bucket1
Requester
```

- Disable the pay-by-requester mode

```
./ ossutil request - payment -- method put oss :// bucket1
BucketOwne r
```

- Query the pay-by-requester configurations

```
./ ossutil request - payment -- method get oss :// bucket1
BucketOwne r
```

```
0 . 178036 ( s ) elapsed
```

Common options

The following table describes the options you can add to the `request - payment` command.

Option	Description
<code>-- method</code>	Specifies the HTTP request method. Valid values: <ul style="list-style-type: none"> · <code>put</code>: enables or disables the pay-by-requester mode. · <code>get</code>: obtains pay-by-requester configurations.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · <code>info</code>: generates prompt logs. · <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79. **: **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.27 rm

The `rm` command is used to delete a bucket, object, or part.

Command syntax

```
./ ossutil rm oss :// bucket [/ prefix ] [- r ] [- b ] [- f ][-  
c file ] [-- include include - pattern ] [-- payer requester ]  
[-- exclude exclude - pattern ]
```

Examples

- Delete an empty bucket

```
./ ossutil rm oss :// bucket1 - b
```



Notice:

- To delete a bucket, you must set the `- b` option.
- Deleted buckets can be re-created by other users, at which point you will no longer be able to use the buckets.

- Clear bucket data and delete the bucket

If the bucket contains data such as objects or parts, you must delete all data stored in the bucket before the bucket can be deleted. The command is as follows:

```
./ ossutil rm oss :// bucket1 - bar
```



Warning:

This command will clear all data in the bucket. Exercise caution when using this command.

- Delete a single object

```
./ ossutil  rm  oss :// bucket1 / path / object
```

- Delete all objects that have a specified prefix

You can add the `- r` option to the `rm` command to recursively delete all objects that have a specified prefix.

```
./ ossutil  rm  oss :// bucket1 / path / - r
```

- Delete multiple objects that meet specified conditions

You can set `-- include /-- exclude` to list objects that meet specified conditions. For more information about `-- include /-- exclude`, see [cp](#).

- Delete all objects except `jpg` objects

```
./ ossutil  rm  oss :// my - bucket / path  -- exclude "*. jpg  
" - r
```

- Delete all objects that contain `abc` in their names and are not in the `jpg` or `txt` format

```
./ ossutil  rm  oss :// my - bucket1 / path  -- include "* abc  
*" -- exclude "*. jpg " -- exclude "*. txt " - r
```

- Delete the upload IDs of incomplete multipart upload tasks initiated to a specified object

You can add the `- m` option to the `rm` command to delete the upload IDs of incomplete multipart upload tasks initiated to a specified object.

```
./ ossutil  rm  - m  oss :// bucket1 / obj1 / test . txt  
Succeed : Total  1  uploadIds .  Removed  1  uploadIds .  
0 . 900715 ( s )  elapsed
```

- Delete the upload IDs of all incomplete multipart upload tasks initiated to objects that have a specified prefix

You can add the `- m` and `- r` options to the `rm` command to delete the upload IDs of all incomplete multipart upload tasks initiated to objects that have a specified prefix.

```
./ ossutil  rm  - m  oss :// bucket1 / ob - r  
Do you really mean to remove recursively multipart  
uploadIds of oss : bucket1 / ob ( y or N )? y  
Succeed : Total  4  uploadIds .  Removed  4  uploadIds .
```

```
1 . 922915 ( s ) elapsed
```

- Delete the upload IDs of all incomplete multipart upload tasks initiated to objects with a specified prefix and the uploaded objects with the prefix

You can add the `-a` and `-r` options to the `rm` command to delete the upload IDs of all incomplete multipart upload tasks initiated to objects with a specified prefix and the uploaded objects with the prefix.

```
./ossutil rm oss :// hello - hangzws - 1 / obj - a - r
Do you really mean to remove recursively objects
and multipart uploadIds of oss :// obj ( y or N )? y
Succeed : Total 1 objects , 3 uploadIds . Removed 1
objects , 3 uploadIds .
```

Common options

The following table describes the options you can add to the `rm` command to delete different content.

Option	Description
<code>-r</code> , <code>--recursive</code>	Recursively performs operations on objects in a bucket. If this option is specified, commands that support this option will perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, the commands will only perform operations on a single specified object.
<code>-b</code> , <code>--bucket</code>	Specifies the bucket on which to perform an operation.
<code>-m</code> , <code>--multipart</code>	Lists only incomplete multipart upload tasks in the bucket. Objects are not displayed.
<code>-a</code> , <code>--all-type</code>	Specifies operations are to be performed on both the objects and incomplete multipart upload tasks in a bucket.
<code>-f</code> , <code>--force</code>	Forces an operation without prompting the user for confirmation.
<code>--encoding-type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code> . If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>--retry-times =</code>	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.

Option	Description
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
-- include	Includes objects that match a specified string, such as *.jpg.
-- exclude	Excludes objects that match a specified string, such as *.txt.
-- payer	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is http://120.79.***:3128 or socks5://120.79.**: **:1080.
-- proxy - user	Specifies the username of the proxy server. The default value is null.
-- proxy - pwd	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.28 set-acl

`set - acl` is used to configure the access control list (ACL) for a bucket or object.

Command syntax

```
./ ossutil set - acl oss :// bucket [/ prefix ] [ acl ] [- r ] [- b ] [- f ] [- c file ]
```

Examples

- Configure the ACL for a bucket

```
./ ossutil set - acl oss :// bucket1 private - b
```

**Note:**

A bucket can have any of the following ACL policies:

- private
- public-read

- **public-read-write**

For more information about ACLs, see [#unique_15](#).

- Configure the ACL for a specified object

```
./ ossutil set - acl oss :// bucket1 / path / object private
```



Note:

An object can have any of the following ACL policies:

- **default:** The object inherits the ACL for the bucket it belongs to.
- **private**
- **public-read**
- **public-read-write**

- Configure the ACL for all objects that have a specified prefix

```
./ ossutil set - acl oss :// bucket1 / path / private - r
```

- Configure the ACL for objects that meet specified conditions

When configuring ACLs, you can use the `-- include /-- exclude` command to select objects that meet specified conditions. For more information, see [cp](#).

- Set the ACL to private for all objects except `jpg` objects

```
./ ossutil set - acl oss :// my - bucket1 / path private --
exclude "*.jpg" - r
```

- Set the ACL to private for all objects that contain `abc` in their names and are not in the `jpg` or `txt` format

```
./ ossutil set - acl oss :// my - bucket1 / path private --
include "* abc *" -- exclude "*.jpg" -- exclude "*.txt" -
r
```

Common options

The following table describes the options you can add to the `set - acl` command to set different ACLs for different objects.

Option	Description
- r , -- recursive	Recursively performs operations on objects in a bucket. If this option is specified, commands that support this option will perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, the commands will only perform operations on a single specified object.
- b , -- bucket	Specifies the bucket on which to perform an operation.
- f , -- force	Forces an operation without prompting the user for confirmation.
-- include	Includes objects that match a specified string, such as *.jpg.
-- exclude	Excludes objects that match a specified string, such as *.txt.
- j , -- jobs	Specifies the number of concurrent tasks when multiple objects are operated. Valid values: 1 to 10000. Default value: 3.
-- encoding - type	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
-- output - dir	Specifies the directory in which output objects are located. Output objects include report objects generated due to errors that occur when you use the cp command to copy multiple objects. For more information about the report objects, see the help information of the cp command. The default value is the ossutil_output directory in the current directory.
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
-- retry - times	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is http://120.79.***:3128 or socks5://120.79. **. **:1080.
-- proxy - user	Specifies the username of the proxy server. The default value is null.

Option	Description
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.

**Note:**

For more information about common options, see [#unique_62](#).

3.3.29 set-meta

The `set - meta` command is used to set object metadata for objects that have been uploaded.

Command syntax

```
./ ossutil set - meta oss :// bucket [/ prefix ] [ header : value
# header : value ...] [-- update ] [-- delete ] [- r ] [- f ] [- c
file ]
```

Examples

- Configure the metadata of a specified object

```
./ ossutil set - meta oss :// bucket1 / path / object x - oss
- object - acl : private
```

If the `-- update` and `-- delete` options are not specified, this command replaces the existing metadata with the specified metadata. If the `[header : value # header : value ...]` option is not specified, the values of headers that cannot be deleted, such as headers without the X-Oss-meta prefix, will remain unchanged. Other metadata will be deleted.

**Note:**

Headers are case-insensitive, whereas values are case-sensitive. The list of headers you can set for objects is as follows:

```
Headers :
Expires ( time . RFC3339 : 2006 - 01 - 02T15 : 04 : 05Z07 :
00 )
X - Oss - Object - Acl
Origin
X - Oss - Storage - Class
Content - Encoding
Cache - Control
Content - Dispositio n
Accept - Encoding
X - Oss - Server - Side - Encryption
Content - Type
```

and headers with the X-Oss-Meta-prefix .

For more information, see [#unique_105](#).

- Configure the metadata of all objects that have a specified prefix

```
./ ossutil set - meta oss :// bucket1 / path / Cache - Control
: no - cache # x - oss - object - acl : private - r
```

If the `- r` option is specified, the metadata of multiple objects with the specified prefix will be configured. When an error occurs during an operation on an object, ossutil records the error information of the object in a report object and then continues to perform the operation on other objects. The information of objects that were processed will not be recorded in the report object. Separate multiple metadata items with number signs (#).

- Update the metadata of a specified object

```
./ ossutil set - meta oss :// bucket1 / path / object x - oss
- object - acl : private -- update
```

The command only updates the specified header of the specified object as the input value. The header value can be null. The other metadata of the specified object remains unchanged. The `-- update` (abbreviated as `- u`) and `-- delete` options cannot be specified together.

- Delete the metadata of a specified object

```
./ ossutil set - meta oss :// bucket1 / obj1 X - Oss - Meta -
delete -- delete
```

If the `-- delete` option is specified, ossutil deletes the specified object header and the value becomes null. The other metadata of the object remains unchanged. This option is not valid for headers that cannot be deleted, such as headers without the X-Oss-Meta prefix. The `-- update` and `-- delete` options cannot be specified together.

- Configure the metadata of multiple objects with specified conditions

When configuring metadata, you can use the `-- include /-- exclude` command to select objects that meet the specified conditions. For more information, see [cp](#).

- Set all objects with a `jpg` extension to the Infrequent Access (IA) storage class

```
./ossutil64 set - meta oss :// my - bucket / path X - Oss -
Storage - Class : IA -- include "*.jpg" - u - r
```

- Set all objects that contain `abc` in their names and are not in the `jpg` or `txt` format to the Standard storage class

```
./ossutil set - meta oss :// my - bucket / path X - Oss -
Storage - Class : Standard -- include "* abc *" -- exclude "*.
jpg" -- exclude "*.txt" - u - r
```

Common options

The following table describes the options that you can add to the `set - meta` command.

Option	Description
<code>- u , -- update</code>	Updates the metadata of a specified object.
<code>-- delete</code>	Deletes the metadata of a specified object.
<code>- r , -- recursive</code>	Recursively performs operations on objects in a bucket. If this option is specified, commands that support this option will perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, the commands will only perform operations on a single specified object.
<code>- f , -- force</code>	Forces an operation without prompting the user for confirmation.
<code>-- include</code>	Includes objects that match a specified string, such as <code>*.jpg</code> .
<code>-- exclude</code>	Excludes objects that match a specified string, such as <code>*.txt</code> .
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code> . If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>- j , -- jobs</code>	Specifies the number of concurrent tasks when multiple objects are operated. Valid values: 1 to 10000. Default value: 3.

Option	Description
- L , -- language	Specifies the language ossutil uses. Valid values: CH and EN. Default value: CH. To set this option to CH, ensure that your system supports UTF-8 encoding.
-- output - dir	Specifies the directory in which output objects are located. Output objects include report objects generated due to errors that occur when you use the <code>cp</code> command to copy multiple objects. For more information about the <code>report</code> objects, see the help information of the <code>cp</code> command. The default value is the <code>ossutil_output</code> directory in the current directory.
-- loglevel	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain corresponding HTTP request and response information.
-- retry - times	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
-- proxy - host	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79.**: **:1080</code> .
-- proxy - user	Specifies the username of the proxy server. The default value is null.
-- proxy - pwd	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.30 sign

The `sign` command is used to generate signed URLs for third-party users to access objects in buckets.



Note:

For more information about object URLs, see [How to obtain the URL of an uploaded object](#).

Command syntax

```
./ ossutil sign oss :// bucket / object [-- timeout time ] [--
traffic - limit limitSpeed ]
```

- `-- timeout` : specifies the timeout value of the object URL. Valid values: non-negative integers. Default value: 60. Unit: seconds.
- `-- traffic - limit` : specifies the bandwidth that is used to access the object over HTTP. Valid values: 819200 to 838860800 (100 KByte/s to 100 MByte/s). Default value: 0 (unlimited). Unit: bit/s.

Examples

- Generate an object URL that has a default timeout value of 60 seconds

```
./ ossutil sign oss :// bucket / path / object
```

- Generate an object URL with a specified timeout value of 3600 seconds

```
./ ossutil sign oss :// bucket / path / object -- timeout
3600
```

- Generate an object URL with the bandwidth of 1 MByte/s

```
./ ossutil sign oss :// bucket / path / object -- traffic -
limit 8388608
```



Note:

Formula for unit conversion: 1 MB = 1,024 KB = 1,048,576 Bytes = 8,388,608 bits

Common options

The following table describes the options you can add to the `sign` command.

Option	Description
<code>-- timeout</code>	Specifies the timeout value of the object URL. Valid values: non-negative integers. Default value: 60. Unit: seconds.
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be url. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.

Option	Description
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> <code>info</code>: generates prompt logs. <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- traffic - limit</code>	Specifies the bandwidth that is used to access the object over HTTP. Valid values: 819200 to 838860800 (100 KByte/s to 100 MByte/s). Default value: 0 (unlimited). Unit: bit/s.



Note:

For more information about common options, see [#unique_62](#).

3.3.31 stat

The `stat` command is used to obtain the description of buckets or objects. For example, you can run the `stat` command to view the object metadata that is set by running the `set - meta` command.

Command syntax

```
./ ossutil stat oss :// bucket [/ object ] [-- encoding - type url ] [-- payer requester ] [- c file ]
```

Examples

- Obtain information of a bucket

```
./ ossutil stat oss :// bucket1
```

- Obtain information of a specified object

```
./ ossutil stat oss :// bucket1 / object
```

- Obtain information of an object whose names contain special characters

`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.

```
./ ossutil stat oss :// bucket1 /% E7 % A4 % BA % E4 % BE % 8B . txt -- encoding - type url
```



Notice:

Bucket names cannot be URL-encoded.

Common options

The following table describes the options you can add to the `stat` command.

Option	Description
<code>-- encoding - type</code>	Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code> . If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.
<code>-- loglevel</code>	Specifies the log level. The default value is <code>null</code> , indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> <code>info</code>: generates prompt logs. <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- retry - times</code>	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
<code>-- payer</code>	Specifies the payer of the request. To enable the pay-by-requester mode, set this option to <code>requester</code> .
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***.3128</code> or <code>socks5://120.79. **. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is <code>null</code> .
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is <code>null</code> .



Note:

For more information about common options, see [#unique_62](#).

3.3.32 update

The `update` command is used to update the `ossutil` version.

Command syntax

```
./ossutil update [-f]
```

This command checks the current and latest versions of `ossutil` and generates their version numbers. If an update is available, this command asks whether you want to

upgrade ossutil to the latest version. If the `-- force` (abbreviated as `- f`) option is specified, the command upgrades ossutil without asking for confirmation when an update is available.

Examples

Upgrade the version of ossutil.

```
./ossutil update
The current version is 1.5.1 and the latest
version is 1.6.0.
Are you sure you want to update the version ( y or
N )? y
Updated .
```

Common options

The following table describes the options you can add to the `update` command.

Option	Description
<code>- f</code> , <code>-- force</code>	Forces an operation without prompting the user for confirmation.
<code>-- retry - times</code>	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> <code>info</code>: generates prompt logs. <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***.***:3128</code> or <code>socks5://120.79.***.***:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.3.33 website

The `website` command is used to add, modify, query, or delete static website hosting and back-to-origin configurations.



Note:

- For more information about static website hosting, see [#unique_110](#).
- For more information about back-to-origin, see [#unique_111](#).

Command syntax

- Add or modify website-related configurations

```
./ ossutil  website  -- method  put  oss :// bucket
local_xml_  file
```

`ossutil` is used to read or write website-related configurations from or to `local_xml_ file` . If the bucket has website-related configurations, new configurations will overwrite the existing configurations.



Note:

The `local_xml_ file` configuration file is in the XML format as follows:

- The `IndexDocument` field is used to specify the default homepage.
- The `ErrorDocument` field is used to specify the default 404 page.
- The `RoutingRules` field is used to configure back-to-origin rules in the redirection or mirroring mode.

You can set the preceding fields as required. The following code provides an example of how to set these fields:

```
<? xml  version =" 1 . 0 "  encoding =" UTF - 8 "? >
< WebsiteCon  figuration >
  < IndexDocum  ent >
    < Suffix > index . html </ Suffix >
  </ IndexDocum  ent >
  < ErrorDocum  ent >
    < Key > error . html </ Key >
  </ ErrorDocum  ent >
  < RoutingRul  es >
    < RoutingRul  e >
      < RuleNumber > 1 </ RuleNumber >
      < Condition >
        < KeyPrefixE  quals > abc </ KeyPrefixE  quals >
        < HttpErrorC  odeReturne  dEquals > 404 </
HttpErrorC  odeReturne  dEquals >
      </ Condition >
      < Redirect >
```

```

        < RedirectType > Mirror </ RedirectType >
        < PassQueryString > true </ PassQueryString >
        < MirrorURL > http://www.test.com </
MirrorURL >
        < MirrorPassQueryString > true </ MirrorPass
QueryString >
        < MirrorFollowRedirect > true </ MirrorFoll
owRedirect >
        < MirrorChecksumMd5 > false </ MirrorChec kMd5 >
        < 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 >
        </ MirrorHead ers >
    </ Redirect >
</ RoutingRule >
< RoutingRule >
    < RuleNumber > 2 </ RuleNumber >
    < Condition >
        < KeyPrefixEquals > abc </ KeyPrefixE quals >
        < HttpStatusCodeReturnedEquals > 404 </
HttpErrorC odeReturne dEquals >
        < IncludeHeader >
        < Key > host </ Key >
        < Equals > test.oss-cn-beijing-internal.
aliyuncs.com </ Equals >
        </ IncludeHea der >
    </ Condition >
    < Redirect >
        < RedirectType > AliCDN </ RedirectType >
        < Protocol > http </ Protocol >
        < HostName > www.test.com </ HostName >
        < PassQueryString > false </ PassQueryString >
        < ReplaceKeyWith > prefix /${key}.suffix </
ReplaceKey With >
        < HttpRedirectCode > 301 </ HttpRedire ctCode >
    </ Redirect >
    </ RoutingRule >
</ RoutingRules >
</ WebsiteCon figuration >

```

- Obtain website-related configurations

```

./ossutil website --method get oss://bucket [
local_xml_file ]

```

The `local_xml_file` parameter specifies the name of the configuration file. If this parameter is specified, website-related configurations will be saved as a local file. If this parameter is not specified, ossutil will display website-related configurations.

- Delete website-related configurations

```
./ ossutil website --method delete oss :// bucket
```

Examples

- Add default homepage and default 404 page configurations for static website hosting

```
./ ossutil website --method put oss :// bucket1 / file / website . xml
```

The content of the `website . xml` file is as follows:

```
<? xml version = " 1 . 0 " encoding = " UTF - 8 "? >
< WebsiteCon figuration >
  < IndexDocum ent >
    < Suffix > test . html </ Suffix >
  </ IndexDocum ent >
  < ErrorDocum ent >
    < Key > errortest . html </ Key >
  </ ErrorDocum ent >
</ WebsiteCon figuration >
```

- Add mirroring-based back-to-origin configurations

```
./ ossutil website --method put oss :// bucket1 / file / website . xml
```

The content of the `website . xml` file is as follows:

```
<? xml version = " 1 . 0 " encoding = " UTF - 8 "? >
< WebsiteCon figuration >
< RoutingRul es >
  < RoutingRul e >
    < RuleNumber > 1 </ RuleNumber >
    < Condition >
      < KeyPrefixE quals > abc </ KeyPrefixE quals >
      < HttpErrorC odeReturne dEquals > 404 </
HttpErrorC odeReturne dEquals >
    </ Condition >
    < Redirect >
      < RedirectTy pe > Mirror </ RedirectTy pe >
      < PassQueryS tring > true </ PassQueryS tring >
      < MirrorURL > http :// www . aliyun . com </
MirrorURL >
      < MirrorPass QueryStrin g > true </ MirrorPass
QueryStrin g >
      < MirrorFoll owRedirect > true </ MirrorFoll
owRedirect >
      < MirrorChec kMd5 > false </ MirrorChec kMd5 >
      < MirrorHead ers >
        < PassAll > true </ PassAll >
        < Pass > myheader1 </ Pass >
        < Pass > myheader2 </ Pass >
        < Remove > myheader3 </ Remove >
        < Remove > myheader4 </ Remove >
        < Set >
```

```

        < Key > myheader5 </ Key >
        < Value > myheader5 </ Value >
    </ Set >
</ MirrorHead ers >
</ Redirect >
</ RoutingRul e >
< RoutingRul es >
</ WebsiteCon figuration >

```

- Obtain website-related configurations

```
./ ossutil website -- method get oss :// bucket1 / file /
website . xml
```

- Delete website-related configurations

```
./ ossutil website -- method delete oss :// bucket1
```

Common options

The following table describes the options you can add to the `website` command.

Option	Description
<code>-- method</code>	Specifies the HTTP request method. Valid values: <ul style="list-style-type: none"> • put: creates or modifies configurations. • get: obtains configurations. • delete: deletes configurations.
<code>-- loglevel</code>	Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: <ul style="list-style-type: none"> • info: generates prompt logs. • debug: generates detailed logs that contain corresponding HTTP request and response information.
<code>-- proxy - host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <code>http://120.79.***.3128</code> or <code>socks5://120.79. **. **:1080</code> .
<code>-- proxy - user</code>	Specifies the username of the proxy server. The default value is null.
<code>-- proxy - pwd</code>	Specifies the password of the proxy server. The default value is null.



Note:

For more information about common options, see [#unique_62](#).

3.4 View all supported options

You can use the `-h` option to view all options supported by `ossutil`.

Command syntax

```
./ossutil -h
```

To view the options supported by a command, run the `ossutil help [command] command`, such as `ossutil help cp`.

Common options

The following section describes some common options that can be used in most `ossutil` commands:

- `-c`, `--config -file`

Specifies the configuration file path of `ossutil`. `ossutil` reads the configuration file during startup, and writes configurations to the file by using the `config` command. When managing buckets that belong to different accounts, you can generate multiple configuration files, and select one to serve as the default configuration file. When managing buckets that belong to other accounts, you can use the `-c` option to specify the corresponding configuration files.

- `-e`, `--endpoint`

Specifies the endpoint of a bucket. When you perform across-region bucket management, you can use the `-e` option to specify the regions.

- `-i`, `--access-key-id`

Specifies the AccessKey ID used to access OSS. When managing buckets that belong to different accounts, you can use the `-i` option to specify the corresponding AccessKey ID.

- `-k`, `--access-key-secret`

Specifies the AccessKey Secret used to access OSS. When managing buckets that belong to different accounts, you can use the `-k` option to specify the corresponding AccessKey Secret.

- -- loglevel

Generates ossutil log files `ossutil . log` in the current working directory. The default value is null, indicating that no log files are generated.

- Valid values:
- info: generates operations logs.

```
./ ossutil [ command ] -- loglevel = info
```

- debug: generates logs that contain HTTP requests and responses and original signature strings to locate problems.

```
./ ossutil [ command ] -- loglevel = debug
```

- -- proxy - host , -- proxy - user , -- proxy - pwd

If your environment requires a proxy server to access websites, you need to use these three options to specify the proxy server information. ossutil uses the specified information to access OSS through the proxy server. Example:

```
./ ossutil ls oss :// bucket1 -- proxy - host http :// 47 .
88 . **. **: 3128 -- proxy - user test -- proxy - pwd test
```

Options

The following table describes all options supported by ossutil.

Option	Description
-s, --short-format	Lists items in short format. The long format is used if this option is not specified.
--bigfile-threshold	Specifies the threshold for enabling the resumable data transfer for large files. Unit: Byte. Valid values: non-negative integers. Default value: 100 MB.
--acl	Configures the access control list (ACL) for an object or bucket.
--range	Specifies the range of an object to be downloaded. The value must be in the 3-9, 3-, or -9 format.
--type	Specifies the calculation type. Valid values: crc64 and md5. Default value: crc64.
-v, --version	Displays the ossutil version and exits.
-u, --update	Updates the ossutil version.
--origin	Specifies the value of the Origin header in an HTTP request.

Option	Description
<code>--upmode</code>	Specifies the upload method. Valid values: normal, append, and multipart. Default value: normal. This option is used with the <code>probe</code> command.
<code>--sse-algorithm</code>	Specifies the server-side encryption algorithm. Valid values : KMS and AES256.
<code>--include</code>	Includes objects that match a specified string, such as <code>*.jpg</code> .
<code>--exclude</code>	Excludes objects that match a specified string, such as <code>*.txt</code> .
<code>-r, --recursive</code>	Recursively performs operations on objects in a bucket. If this option is specified, commands that support this option will perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, the commands will only perform operations on the specified object.
<code>--addr</code>	Specifies a network address, which is a domain name in most cases. This option is used with the <code>probe</code> command.
<code>--kms-masterkey-id</code>	Specifies the CMK ID used for encryption in Key Management Service (KMS).
<code>-m, --multipart</code>	Specifies that operations are only to be performed on uncompleted multipart upload tasks in a bucket instead of objects.
<code>-d, --directory</code>	Lists files and subdirectories in the current directory, instead of recursively displaying all objects in all subdirectories.
<code>--payer</code>	Specifies the payer of the request. You can set this option to requester to enable the pay-by-requester mode.
<code>--maxuploadspeed</code>	Specifies the maximum upload speed. Unit: KByte/s. Default value: 0 (unlimited).
<code>--retry-times</code>	Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.
<code>-c, --config-file</code>	Specifies the configuration file path of ossutil. ossutil reads the configuration file during startup, and writes configurations to the file by using the <code>config</code> command.
<code>--download</code>	Downloads an object from OSS. This option is used with the <code>probe</code> command.

Option	Description
<code>-j, --jobs</code>	Specifies the number of concurrent operations performed on multiple objects. Valid values: 1 to 10000. Default value: 3.
<code>-a, --all-type</code>	Specifies that operations are to be performed on both the objects and uncompleted multipart upload tasks in a bucket.
<code>--disable-empty-referer</code>	Indicates that the referer field cannot be left empty. This option is used with the <code>referer</code> command.
<code>--method</code>	Specifies the HTTP request method, which can be PUT, GET, or DELETE.
<code>--output-dir</code>	Specifies the destination to write output files. Output files include report files generated due to errors that occur when you use the <code>cp</code> command to copy multiple files at a time. To obtain more information about the report files, use the help option with the <code>cp</code> command. The default value is the <code>ossutil_output</code> directory in the current directory.
<code>--meta</code>	Configures the metadata of an object in the <code>[header:value#header:value...]</code> format. Example: <code>Cache-Control: no-cache # Content-Encoding: gzip</code> .
<code>--object</code>	Specifies the name of an object in OSS. This option is used with the <code>probe</code> command.
<code>-e, --endpoint</code>	Specifies the basic endpoint of ossutil. This option value will overwrite the corresponding endpoint in the configuration file. Note that this value must be a second-level domain name.
<code>--limited-num</code>	Specifies the number of buckets displayed on each page.
<code>-L, --language</code>	Specifies the language of ossutil. Valid values: CH and EN. Default value: CH. If you plan to set this option to CH, ensure that your system supports UTF-8 encoding.
<code>--delete</code>	Specifies a delete operation.
<code>-b, --bucket</code>	Specifies the bucket on which to perform the operation.
<code>--disable-crc64</code>	Disables CRC64. By default, ossutil enables CRC64 during data transmission.
<code>--upload</code>	Uploads an object to OSS. This option is used with the <code>probe</code> command.

Option	Description
<code>--part-size</code>	Specifies the part size in Byte. By default, ossutil calculates the appropriate part size based on the object size. If you need to optimize performance or are operating under special constraints, you can set this option to any positive integer.
<code>--timeout</code>	Specifies the timeout period for a signed URL request. Unit: seconds. Valid values: non-negative integers. Default value: 60.
<code>-k, --access-key-secret</code>	Specifies the AccessKey Secret used to access OSS. This option value will overwrite the corresponding configurations in the configuration file.
<code>--checkpoint-dir</code>	Specifies the checkpoint directory path. Default value: <code>.ossutil_checkpoint</code> . If a resumable data transfer fails, ossutil automatically creates this directory and records the checkpoint information in the directory. If a resumable data transfer succeeds, ossutil deletes this directory. If this option is specified, ensure that you have permissions to delete the specified directory.
<code>--url</code>	Specifies the URL of an object. This option is used with the <code>probe</code> command.
<code>--marker</code>	Specifies the bucket name, object name, or multipart upload ID from which you want to start listing.
<code>-f, --force</code>	Forces an operation without prompting the user for confirmation.

Option	Description
--snapshot-path	<p>Specifies the directory used in certain scenarios to accelerate the incremental upload or download of multiple objects. If you use this option to upload or download objects, ossutil takes a snapshot of the upload or download and stores it in a specified directory. Then, the next time objects are uploaded or downloaded, ossutil will read the snapshot information from the specified directory to perform incremental upload or download. The snapshot directory you specified must be a writable directory in your local file system. If the directory does not exist, ossutil creates a directory to store the snapshot information. If the directory already exists, ossutil reads and updates the snapshot information in the directory, and incrementally uploads or downloads objects that previously failed to be uploaded or downloaded and have been modified.</p> <p>Note: If you use this option, the recorded local lastModifiedTime of previously uploaded or downloaded objects is compared with that of objects pending upload or download to determine whether to skip these objects. If you use this option to upload an object, make sure that no other users update the object in OSS between two upload operations. In other scenarios where objects are updated in OSS between two upload operations, use the --update option to incrementally upload or download the objects. ossutil does not automatically delete the snapshot information in the directory specified by snapshot-path. If you no longer need the snapshot information, delete the directory.</p>
--loglevel	<p>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</p> <ul style="list-style-type: none"> · info: generates prompt logs. · debug: generates detailed logs that contain HTTP requests and responses.
--storage-class	<p>Specifies the storage method of an object. Valid values: Standard, IA, and Archive. Default value: Standard.</p>
-i, --access-key-id	<p>Specifies the AccessKey ID used to access OSS. This option value will overwrite the corresponding configurations in the configuration file.</p>
-t, --sts-token	<p>Optional. This option specifies the STS token used to access OSS. This option value will overwrite the corresponding configurations in the configuration file.</p>

Option	Description
--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 to download. The value of this option is in the "partition number: total number of partitions" format. A value of 1:5 indicates that ossutil downloads partition 1 out of the total five partitions. Partitioning rules for objects are determined by ossutil. This option divides an object to be downloaded into multiple partitions that can be downloaded by multiple ossutil commands. Each ossutil command downloads its own partition. You can run multiple ossutil commands on different machines at the same time.
--bucketname	Specifies the name of a bucket. This option is used with the <code>probe</code> .
--encoding-type	<p>Specifies the encoding type of the object name. If this option is specified, this value must be <code>url</code>. If this option is not specified, the object name is not encoded. Bucket names cannot be URL-encoded.</p> <div data-bbox="576 1178 1433 1462" style="background-color: #f0f0f0; padding: 10px;">  Notice: The value of this option must be in the <code>oss://bucket/url_encode (object)</code> format. For example, the <code>cloud_url oss://bucket/object</code> must be entered as <code>oss://bucket/url_encode (object)</code>. The <code>oss://bucket/</code> string does not need to be encoded. </div>
--origin	Specifies the value of the Origin header in an HTTP request . This option value indicates the source domain of a cross-origin request.
--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 values of the Access-Control-Request-Headers header in an HTTP request. This option value indicates header fields, except simple ones, to be used in an actual request. To specify multiple headers, separate different headers with commas (,) and enclose the headers with double quotation marks ("). Example: <code>--acr-headers "header1, header2, header3"</code> .

Option	Description
<code>--upload-id-marker</code>	Specifies the multipart upload ID from which you want to start listing.
<code>-h, --help</code>	Displays help information for a specified command.
<code>--proxy-host</code>	Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 proxies are supported. An example of the URL is <code>http://120.79.***:3128</code> or <code>socks5://120.79.**.**:1080</code> .
<code>--proxy-user</code>	Specifies the username of the proxy server. The default value is null.
<code>--proxy-pwd</code>	Specifies the password of the proxy server. The default value is null.
<code>--traffic-limit</code>	Specifies the bandwidth that is used to access the object over HTTP. Unit: bit/s. Valid values: 819200 to 838860800 (100 KByte/s to 100 MByte/s). Default value: 0 (unlimited). This option is used with the <code>sign</code> command.

3.5 Bucket-related commands

This topic describes how to use the Alibaba Cloud OSS open-source tool `ossutil` to run bucket-related commands. Specifically, you create, delete, or list a bucket, or change the ACL of a bucket. You can also use this tool to manage bucket-related items such as objects, uncompleted multipart upload tasks, to manage Cross-Origin Resource Sharing (CORS) rules, log rules, or anti-leech rules, or to troubleshoot the OSS network.



Note:

- Before you can run bucket-related commands, you must first upgrade your `ossutil` to the latest version and run the `config` command to configure your AccessKey. For more information, see [Quick start](#).
- Bucket management functions besides those described here are not supported by `ossutil`. If you require such functions, use the `osscmd` tool. For more information, see [Quick installation](#).

Create a bucket

- Create a bucket

```
./ ossutil mb oss :// bucketname [-- acl = ACL ][-- storage -  
class sc ][- c file ]
```

If the bucket is created, ossutil prints the interval of time needed to create the bucket and exits. If the bucket failed to be created, ossutil outputs the corresponding error information.



Note:

For information about how to use the `mb` command, run the `ossutil help mb` command.

- Create a bucket and set its ACL

You can use the `-- acl` parameter to set the ACL for a bucket. The default ACL is `private`. The following are available ACLs:

- `private` : Anonymous users are not allowed to read from or write to objects in the bucket. A signature is required for access.
- `public - read` : Anonymous users are allowed only to read from objects in the bucket.
- `public - read - write` : Anonymous users are allowed to read from and write to objects in the bucket.



Note:

For more information on access control, see [Access control based on ACLs](#).

For example, run the following command to create a bucket and set its ACL to `public - read - write` :

```
./ ossutil mb oss :// bucket -- acl = public - read - write
```

- Create a bucket and set its storage class

You can use the `-- storage - class` parameter to set the storage class of a bucket. The default storage class is `Standard` . The following are available storage classes:

- `Standard`
- `Infrequent Access`
- `Archive`



Note:

For more information on storage classes, see [Overview](#).

For example, run the following command to create a bucket and set its storage class to `Infrequent Access` :

```
./ ossutil mb oss :// bucket -- storage - class IA
```

Change the ACL for a bucket

You can run the `set - acl` command to change the ACL for a bucket. In this command, you must set the `- b` parameter.

For example, run the following command to change the ACL for a bucket to `private` :

```
./ ossutil set - acl oss :// bucket1 private - b
```



Note:

For information about how to use the `set - acl` command, run the `ossutil help set - acl` command.

Delete a bucket

- Delete an empty bucket

```
./ ossutil  rm  oss :// bucket  - b
```



Notice:

- You must set the `- b` parameter when you delete a bucket.
- The bucket you delete may be re-created by another user. However, in such case, you will no longer own this bucket.
- For information about how to use the `rm` command, run the `ossutil help rm` command.

- Clear and delete a bucket

If a bucket contains object or multipart data, you must first delete the object or multipart data before you delete the bucket.

```
./ ossutil  rm  oss :// bucket  - bar
```



Warning:

If you run the preceding command, all the data in your bucket is deleted.

List buckets

- List all your buckets

You can run one of the following two commands to list all your buckets:

```
- ./ ossutil  ls
```

```
- ./ ossutil  ls  oss ://
```



Note:

The `- s` parameter is used to list your buckets in a simple structure. For information about how to use the `ls` command, run the `ossutil help ls` command.

Example:

```
./ ossutil  ls
CreationTi me          Region
StorageCla ss      BucketName
```

```

2016 - 10 - 2116 : 18 : 37 + 0800 CST oss - cn -
hangzhou Archive oss :// go - sdk - test - bucket -
xyz - for - object
2016 - 12 - 0115 : 06 : 21 + 0800 CST oss - cn -
hangzhou Standard oss :// ossutil - test
2016 - 07 - 1817 : 54 : 49 + 0800 CST oss - cn -
hangzhou Standard oss :// ossutilcon fig
2016 - 07 - 2010 : 36 : 24 + 0800 CST oss - cn -
hangzhou IA oss :// ossutilupd ate
2016 - 11 - 1413 : 08 : 36 + 0800 CST oss - cn -
hangzhou IA oss :// yyyyyy
2016 - 08 - 2509 : 06 : 10 + 0800 CST oss - cn -
hangzhou Archive oss :// ztzt
2016 - 11 - 2121 : 18 : 39 + 0800 CST oss - cn -
hangzhou Archive oss :// ztztzt
Bucket Number is : 7
0 . 252174 ( s ) elapsed

```

- **List your buckets by page**

```
./ ossutil ls oss :// -- limited - num = ${ num } -- marker = ${
bucketname }
```

If you created a large number of buckets, you can use the `-- limited - num` and `-- marker` parameters to list your buckets by page.

- `-- limited - num` : the number of buckets displayed on each page.
- `-- marker` : the name of the bucket from which ossutil starts to list your buckets. Your buckets are sorted alphabetically and displayed by page. In most cases, ossutil lists your buckets starting from the bucket queried and displayed on the previous page.

For example, run the following command to list the first two buckets by page:

```

./ ossutil ls oss :// -- limited - num = 1 - s
oss :// bucket1
Bucket Number is : 1
0 . 303869 ( s ) elapsed
$ ./ ossutil ls oss :// -- limited - num = 1 - s -- marker =
bucket1
oss :// bucket2
Bucket Number is : 1
0 . 257636 ( s ) elapsed

```

List objects

- **List all the objects in a bucket**

```
./ ossutil ls oss :// bucketname
```

Example:

```
./ ossutil ls oss :// ossutil - test
```

```

LastModifiedTime      Size ( B )  StorageClass
ETAG                  ObjectName
2016 - 12 - 0115 : 06 : 37 + 0800  CST          10363812
Standard              61DE142E5A FF9A674870  7D4A77BFBC  FB          oss
:// ossutil - test / a1
2016 - 12 - 0115 : 06 : 42 + 0800  CST          10363812
Standard              61DE142E5A FF9A674870  7D4A77BFBC  FB          oss
:// ossutil - test / a2
2016 - 12 - 0115 : 06 : 45 + 0800  CST          10363812
Standard              61DE142E5A FF9A674870  7D4A77BFBC  FB          oss
:// ossutil - test / a3
Object Number is : 3
0 . 007379 ( s ) elapsed

```

- List all your objects and uncompleted multipart upload tasks

```
./ ossutil ls oss :// bucket - a
```

Example:

```

./ ossutil ls oss :// bucket1 - a
LastModifiedTime      Size ( B )  StorageClass
ETAG                  ObjectName
2015 - 06 - 0514 : 06 : 29 + 0000  CST          201933
Standard              7E2F4A7F1A C9D2F0996E  8332D5EA5B  41          oss
:// bucket1 / dir1 / obj11
2015 - 06 - 0514 : 36 : 21 + 0000  CST          201933
Standard              6185CA2E8E B8510A61B3  A845EAFE41  74          oss
:// bucket1 / obj1
2016 - 04 - 0814 : 50 : 47 + 0000  CST          6476984
Standard              4F16FDAE7A C404CEC8B7  27FCC67779  D6          oss
:// bucket1 / sample . txt
Object Number is : 3
InitiatedTime        UploadID
ObjectName
2017 - 01 - 1303 : 45 : 26 + 0000  CST          15754AF798
0C4DFB8193 F190837520 BB          oss :// bucket1 / obj1
2017 - 01 - 1303 : 43 : 13 + 0000  CST          2A1F9B4A95
E341BD9285 CC42BB950E E0          oss :// bucket1 / obj1
2017 - 01 - 1303 : 45 : 25 + 0000  CST          3998971ACA
F94AD9AC48 EAC1988BE8 63          oss :// bucket1 / obj2
2017 - 01 - 2011 : 16 : 21 + 0800  CST          A20157A7B2
FEC4670626 DAE0F4C007 3C          oss :// bucket1 / tobj
UploadId Number is : 4
0 . 191289 ( s ) elapsed

```

- List all your objects by page

```
./ ossutil ls oss :// bucket -- limited - num = ${ num } --
marker = ${ obj }
```

You can use the `-- limited - num` and `-- marker` parameters to list your objects by page. This is similar to listing your buckets by page in [Bucket-related commands](#).

```
./ ossutil ls oss :// ossutil - test -- limited - num = 1
```

```

LastModifiedTime          Size ( B )  StorageClass
ETAG                      ObjectName
2016 - 12 - 0115 : 06 : 37 + 0800  CST        10363812
Standard 61DE142E5A FF9A674870 7D4A77BFBC FB          oss
:// ossutil - test / a1
Object Number is : 1
0 . 007379 ( s ) elapsed
$./ ossutil ls oss :// ossutil - test -- limited - num = 1 --
marker = a1
LastModifiedTime          Size ( B )  StorageClass
ETAG                      ObjectName
2016 - 12 - 0115 : 06 : 42 + 0800  CST        10363812
Standard 61DE142E5A FF9A674870 7D4A77BFBC FB          oss
:// ossutil - test / a2
Object Number is : 1
0 . 008392 ( s ) elapsed

```

- List your objects in a simple structure

```
./ ossutil ls oss :// bucket - s
```

Example:

```

./ ossutil ls oss :// ossutil - test
oss :// ossutil - test / a1
oss :// ossutil - test / a2
oss :// ossutil - test / a3
Object Number is : 3
0 . 007379 ( s ) elapsed

```

- List your objects in a simulated directory structure

```
./ ossutil ls oss :// bucket - d
```

If you do not want to list all the objects recursively in the subdirectories of your current directory, you can use the `-d` parameter to list the objects and subdirectories.

Example:

```

./ ossutil ls oss :// bucket1 - s - d
oss :// bucket1 / obj1
oss :// bucket1 / sample . txt
oss :// bucket1 / dir1 /
Object and Directory Number is : 3
0 . 119884 ( s ) elapsed

```

- List all the objects in a bucket to which a domain name is attached

For more information, see [Object-related commands](#).

List uncompleted multipart upload tasks and relevant information

- List your uncompleted multipart upload tasks

```
./ ossutil ls oss :// bucket - m
```

You can use the `- m` parameter to list the uncompleted multipart upload tasks in your current bucket.

Example:

```
./ ossutil ls oss :// bucket1 - m
InitiatedT ime UploadID
ObjectName
2017 - 01 - 1303 : 45 : 26 + 0000 CST 15754AF798 0C4DFB8193
F190837520 BB oss :// bucket1 / obj1
2017 - 01 - 1303 : 45 : 25 + 0000 CST 3998971ACA F94AD9AC48
EAC1988BE8 63 oss :// bucket1 / obj2
2017 - 01 - 2011 : 16 : 21 + 0800 CST A20157A7B2 FEC4670626
DAE0F4C007 3C oss :// bucket1 / tobj
UploadID Number is : 30 . 009424 ( s ) elapsed
```

- List the parts to be uploaded for all your objects

```
./ ossutil getallpart size oss :// bucket
```

- List the parts to be uploaded for a specified object

```
./ ossutil listpart oss :// bucket / object uploadid
```

The `uploadid` parameter specifies the upload task ID of an object whose parts are to be uploaded.



Note:

For more information about multipart upload, see [Multipart-related commands](#).

Configure bucket tagging

By setting the value of `method` to `put`, `get`, or `delete` when running the `bucket - tagging` command, you can add tags to a bucket or modify, query, or delete the tags added to a bucket. For more information about bucket tagging, see [Bucket tagging](#).



Note:

- Only the bucket owner and authorized RAM users can add tags to a bucket. Otherwise, the 403 Forbidden error is returned with the error code `AccessDenied`.
- The key and value of a tag must be separated by a number sign (#).
- You can add a maximum of 20 tags (key-value pairs) to a bucket. Tags must be separated by spaces.
- A tag can contain letters, numbers, spaces, and the following symbols:
+ - = . _ : /
- The maximum length of a key is 64 bytes. The key of a tag cannot be null or be prefixed with `http ://`, `https ://`, or `Aliyun`.
- The maximum length of a tag value is 128 bytes. The value of a tag can be null.
- The key and value of a tag must be UTF-8 encoded.
- We recommend that you run the `ossutil help bucket - tagging` command for more information before running the `bucket - tagging` command.

- You can run the following command to add tags to a bucket or modify the tags added to a bucket:

```
./ ossutil bucket - tagging -- method put oss :// bucket
tagkey1 # tagvalue1 tagkey2 # tagvalue2
```

The preceding command adds specified tags to the bucket if no tag is added to the bucket and overwrites the existing tags if the bucket already has tags.

- You can run the following command to query the tags added to a bucket:

```
./ ossutil bucket - tagging -- method get oss :// bucket
```

- You can run the following command to delete the tags added to a bucket:

```
./ ossutil bucket - tagging -- method delete oss :// bucket
```

Configure bucket encryption

You can set the `method` parameter to `put`, `get`, or `delete` in the `bucket - encryption` command to add, modify, query, or delete the encryption settings of a bucket. For more information about bucket encryption, see [Server-side encryption](#).

- You can run the following command to add the encryption settings for a bucket:

```
./ ossutil bucket - encryption -- method put oss :// bucket
-- sse - algorithm algorithmN ame [-- kms - masterkey - id
keyid ]
```

- `-- sse - algorithm` : Specifies the encryption method for the bucket, which can be set to KMS or AES256.
- `-- kms - masterkey - id` : Specifies the CMK ID used for encryption. Set this parameter as required if the value of `-- sse - algorithm` is KMS. This parameter cannot be set if the value of `-- sse - algorithm` is AES256.

The preceding command sets an encryption method for a bucket if the encryption method is not set for the bucket and modifies the encryption settings for the bucket if the encryption method is set for the bucket.

Example:

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

- You can run the following code to obtain the encryption settings for a bucket:

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

- You can run the following code to delete the encryption settings for a bucket:

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

Manage Cross-Origin Resource Sharing rules

You can set the `method` parameter in the `cors` command to `put` , `get` , or `delete` to add, change, query, or delete the CORS rule of a bucket. For more information, see [Set CORS rules](#).



Note:

For information about how to use the `cors` command, run the `ossutil help cors` command.

- Add or change the CORS rule of a bucket

```
./ossutil cors --method put oss://bucket local_xml_file
```

Ossutil reads CORS rules from the `local_xml_file` configuration file. If no CORS rule is set for your bucket, ossutil adds the corresponding CORS rule obtained from the configuration file to your bucket. If a CORS rule is set for your bucket, ossutil changes this CORS rule to the CORS rule that is obtained from the configuration file.

**Note:**

The `local_xml_file` configuration file is in XML format as follows:

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

- Obtain the CORS rule of a bucket

```
./ossutil cors --method get oss://bucket [local_xml_file]
```

If the `local_xml_file` parameter is set, ossutil saves the obtained CORS rule to the `local_xml_file` configuration file on your computer. If this parameter is null, ossutil displays the obtained CORS rule on your screen.

- Delete the CORS rule of a bucket

```
./ossutil cors --method delete oss://bucket
```

Manage log rules

You can set the `method` parameter in the logging command to `put`, `get`, or `delete` to add, change, query, or delete the log rule of a bucket. For more information, see [Set access logging](#).

**Note:**

For information about how to use the `cors` command, run the `ossutil help logging` command.

- Add or change the log rule of a bucket

```
./ ossutil logging -- method put oss :// bucket oss ://  
target - bucket /[ prefix ]
```

If log management is disabled, run this command to save your bucket access logs as objects to the bucket specified by the `target - bucket` parameter. However, if log management is enabled, you can run this command to change the directory for storing your bucket access logs.

The `prefix` parameter specifies the directory and prefix for storing your bucket access logs. If this parameter is set, ossutil saves your bucket access logs to the specified directory in the bucket specified by the `target - bucket` parameter. If this parameter is null, ossutil saves your bucket access logs to the root directory in the bucket specified by the `target - bucket` parameter. For log object naming conventions, see [Set logging](#).

- Obtain the log rule of a bucket

```
./ ossutil logging -- method get oss :// bucket [  
local_xml_ file ]
```

If the `local_xml_ file` parameter is set, ossutil saves the obtained log rule to the `local_xml_ file` configuration file on your computer. If this parameter is null, ossutil displays the obtained log rule on your screen.

- Delete the log rule of a bucket

```
./ ossutil logging -- method delete oss :// bucket
```

Manage anti-leech rules

You can set the `method` parameter in the `referer` command to `put`, `get`, or `delete` to add, change, query, or delete the anti-leech rule of a bucket. For more information, see [Anti-leech settings](#).



Note:

For information about how to use the `referer` command, run the `ossutil help referer` command.

- Add or change the anti-leech rule of a bucket

```
./ ossutil referer -- method put oss :// bucket referer -  
value [-- disable - empty - referer ]
```

If no anti-leech rule is set for the bucket, ossutil adds the specified anti-leech rule. If an anti-leech rule already exists, ossutil changes this anti-leech rule to the specified anti-leech rule.

- `referer - value` : enables and specifies the Referer whitelist that includes a list of domain names separated by spaces. The whitelist can contain wildcard characters (*) and question marks (?). Only the OSS access requests from the domains included in the whitelist are permitted.
- `-- disable - empty - referer` : specifies whether the Referer field can be left unspecified. If the `-- disable - empty - referer` parameter is used in the `referer` command, the Referer field cannot be left unspecified and only the OSS access requests whose HTTP or HTTPS headers contain this field are permitted. If the `-- disable - empty - referer` parameter is not used, the Referer field can be left unspecified.

For example, run the following command to set the anti-leech rule for a bucket while disallowing the Referer field to be left unspecified:

```
./ ossutil referer -- method put oss :// ossutil - test www  
. test1 . com www . test2 . com -- disable - empty - referer
```

- Obtain the anti-leech rule of a bucket

```
./ ossutil referer -- method get oss :// bucket [  
local_xml_ file ]
```

If the `local_xml_ file` parameter is set, ossutil saves the obtained anti-leech rule to the `local_xml_ file` configuration file on your computer. If this parameter is null, ossutil displays the obtained anti-leech rule on your screen.

- Delete the anti-leech rule of a bucket

```
./ ossutil referer -- method delete oss :// bucket
```

Troubleshoot OSS network

After you run the `probe` command, ossutil prompts you with possible causes to upload and download failures. This may include OSS network faults or inappropriate settings to basic parameters.

**Note:**

For information about how to use the `probe` command, run the `ossutil help probe` command.

- Download an object from a bucket by using the object URL and output a troubleshooting report

```
./ossutil probe --download --url http_url [--addr = domain_name] [file_name]
```

After downloading an object from a bucket to your computer by using the object URL, you can test your network transmission quality and output a troubleshooting report.

- `--url` : the URL of an object in the bucket.

- If the ACL of the object is `public-read`, the URL does not carry a signature, for example, `https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg`.

- If the ACL of the object is `private`, the URL carries a signature and starts and ends with a double quotation mark ("), for example, "`https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg?Expires=1552015472&OSSAccessKeyId=TMPxxxxxxxx5r9f1FV12y8_Qis6LUVmvoSCUSs7aboCCHtydQ0axN32Sn-UvyY3AAAawLAIUarYNLc087AKMEcE503AxxxxxocFAQURdZYyVFyq0W8QkGAN-bamUiQ&Signature=bIa4llbMbl drl7rwckr%2FXXvTtxw%3D`".

**Note:**

For information about how to obtain the URL of an object, see [How to get the link address of the object in OSS](#).

- `--addr = domain_name` : the domain or IP address to which the `ping` command is initiated while the object is being downloaded. This parameter

is optional. If you do not use this parameter, ossutil does not probe another domain or IP address.

- If you use the default parameter value, ossutil runs the `ping` command to check whether communications between your OSS network and `www.aliyun.com` are normal.

- If you specify a domain name or IP address, ossutil runs the `ping` command to check whether communications between your OSS network and the domain or IP address are normal.

- `file_name` : the directory for storing the downloaded object. This parameter is optional. If you do not use this parameter, ossutil saves the downloaded object to the current directory and determines the object name. If you use this parameter to specify an object or directory name, ossutil names the downloaded object by using the specified object name or saves the downloaded object to the specified directory.

- Download an object from a bucket and output a troubleshooting report

```
./ ossutil probe -- download -- bucketname bucket - name [--
object = object_name ]
[-- addr = domain_name ] [ file_name ]
```

- `-- bucketname` : the name of the bucket from which the object is downloaded.
- `-- object` : the directory where the downloaded object is stored. This parameter is optional. If you do not use this parameter, ossutil generates a temporary object, uploads it to the bucket specified by the `bucket - name` parameter, and then downloads this object. After this object is downloaded, ossutil deletes it from your local computer and bucket.

- Check the upload result and output a troubleshooting report

```
./ ossutil probe -- upload [ file_name ] -- bucketname bucket
- name [-- object = object_name ] [-- addr = domain_name ] [-- upmode ]
```

- `file_name` : the name of the object that you want to upload to the bucket specified by the `bucket - name` parameter. The `file_name` parameter is optional. If you do not use this parameter, ossutil generates a temporary object

and uploads it to the specified bucket. After the probing is completed, ossutil deletes this temporary object.

- `-- object` =: the name of an object or directory. This parameter is optional. An example parameter value is `path / myphoto . jpg` , which specifies the object name after the object is uploaded. If you do not use this parameter, ossutil generates a name for the uploaded object. After the probing is completed, ossutil deletes this object.
- `-- upmode` : the upload method. This parameter is optional. The default parameter value is `normal` . The following are available upload methods:
 - `normal`
 - `append`
 - `multipart`

- Obtain a troubleshooting report

After running the `probe` command, you can view each task execution step and the overall upload or download result.

- If a multiplication sign (\times) appears following a step, then this step failed. If a multiplication sign (\times) does not appear, this step succeeded.
- If the upload or download succeeded, ossutil outputs the object size and the time at which the object was uploaded or downloaded. If the upload or download failed, ossutil outputs the failure cause or troubleshooting advice.



Note:

Ossutil may not output troubleshooting advice for some errors. In this case, you can troubleshoot the problems based on the error codes by following the instructions provided in [Exception handling](#).

After running the `probe` command, ossutil generates an object whose name starts with `probe` in your current directory. This object contains details about the commands that you have run to troubleshoot problems.

3.6 Object-related commands

Ossutil allows you to upload/download/copy a file, set the ACL and meta of an object, and view the meta information of an object.

Run the config command to configure the AccessKey pair before running these commands.

- Upload/Download/Copy a file

You are strongly advised to use `ossutil help cp` to view the help information before running the `cp` command.

When running the `cp` command to upload/download/copy a file, use the `-r` option to copy a folder. Ossutil implements multipart upload by default for large files and supports resumable data transfers (the threshold of large files for which multipart upload is enabled can be set using the `--bigfile - threshold` option.)

Use the `-f` option to forcibly upload a file by default. If a file exists with the same name on the target end, the file is overwritten directly.

If an error occurs to a file during file uploading/downloading/copying in batches, ossutil logs the error information in the report file by default, skips this file, and performs operations on other files. For more information, see `ossutil help cp`.



Note:

Ossutil does not continue to copy other files if the bucket does not exist, or if the permission verification result is invalid due to an incorrect AccessKeyID or AccessKeySecret.

Ossutil supports the incremental uploading policies `--update` and `--snapshot - path` in specific scenarios. For more information, see `ossutil help cp`.

From ossutil 1.0.0.Beta1, `crc64` is enabled by default during file uploading.

- Upload a single file:

```

$./ ossutil cp a oss :// ossutil - test
Succeed : Total num : 1 , size : 230 . OK num : 1 (
upload 1 files ).
0 . 699795 ( s ) elapsed

```

- Upload a folder:

```

$./ ossutil cp - r dir oss :// ossutil - test
Succeed : Total num : 35 , size : 464 , 606 . OK num :
35 ( upload 34 files , 1 directorie s ).

```

0 . 896320 (s) elapsed

- **Modify the storage class of an object**



Notice:

To modify the storage class of an object that is less than or equal to 5 GB, run the `set - meta` command. To modify the storage class of an object that is greater than 5 GB, run the `cp` command.

- You can run the `set - meta` command to modify the storage class of an object.

■ Run the following command to set the storage class of a single object to IA:

```
./ossutil set - meta oss :// hello - hangzws / 0104_6 .
jpg X - Oss - Storage - Class : IA - u
```

■ Run the following command to set the storage class of all objects in a folder to Standard:

```
./ossutil set - meta oss :// hello - hangzws / abc / X -
Oss - Storage - Class : Standard - r - u
```

- You can run the `cp` command to upload an object while using the `-- meta` option to modify the storage class of the object.

■ Run the following command to upload a single file and set the storage class of the file to IA:

```
ossutil cp ~/ Downloads / sys . log oss :// hello -
hangzws / test / -- meta X - oss - Storage - Class : IA
```

■ Run the following command to upload a folder and set the storage class of all files in the folder to IA:

```
./ossutil cp ~/ libs3 / oss :// hello - hangzws / test /
-- meta X - oss - Storage - Class : IA - r
```

■ Run the following command to modify the storage class of an existing object to Archive:

```
./ossutil cp oss :// hello - hangzws / 0104_6 . jpg
oss :// hello - hangzws / 0104_6 . jpg -- meta X - oss -
Storage - Class : Archive
```

■ Run the following command to modify the storage class of all objects in an existing folder to Standard:

```
./ossutil cp oss :// hello - hangzws / test / oss ://
hello - hangzws / test / -- meta X - oss - Storage - Class :
Standard - r
```



Notice:

- The storage class of an object cannot be changed from Archive to other classes by running the `set - meta` or `cp` command. You must run the `restore` command first to change the storage class of the object to IA, and then run the `set - meta` or `cp` command to change the storage class of the object to other classes.
- An object is considered as overwritten and may be charged when you run the `cp` command to overwrite the object. An object of the IA or Archive class will be charged if it is overwritten within 30 and 60 days separately after it is created. For example, if you change the storage class of an object from IA to Archive or Standard by running the `cp` command 10 days after the object is created, early deletion fees for 20 days will be charged.

- Performance tuning for uploading, downloading, or copying a file

In the `cp` command, the `-- jobs` and `-- parallel` options are used to control the number of concurrent operations. If the default number of concurrent operations set by ossutil cannot meet the performance requirement, you can modify the values of the two options to adjust the performance.

- The `-- jobs` option controls the number of concurrent operations enabled among files when multiple files are uploaded, downloaded, or copied.
- The `-- parallel` option controls the number of concurrent operations enabled for a large file when the large file is uploaded/downloaded/copied in the multipart method.

By default, ossutil calculates the number of concurrent operations based on the file size (this option does not work for small files, and the threshold for large files to be uploaded/downloaded/copied in the multipart method can be controlled by the `-- bigfile - threshold` option). When large files are uploaded/downloaded/copied in batches, the actual number of concurrent operations is calculated by multiplying the number of jobs by the number of parallel operations.



Warning:

- We recommend that you adjust the number of concurrent operations to a value smaller than 100 if the resources of your ECS instance or server (such as network bandwidth, memory, and CPU) are limited. If the resources are not fully occupied, you can try to set the concurrent operations to a larger value.

- If there are too many concurrent operations, the uploading, downloading, and copying performance of ossutil may degrade, or an EOF error may occur due to inter-thread resource switching and snatching. To resolve this issue, you must adjust the values of `-- jobs` and `-- parallel` based on the actual conditions. To perform pressure testing, set the two options to small values first, and slowly adjust them to the optimal values.

- Configure the ACL of an object

Ossutil uses the `set-acl` command to configure the ACL of an object. You can use the `-r` option to configure the ACLs of objects in batches.

For more information, see `ossutil help set-acl`.

```
$. /ossutil set -acl oss :// dest / a private
0 . 074507 ( s ) elapsed
```

Configure the ACLs of objects in batches:

```
$. /ossutil set -acl oss :// dest / a private -r
Do you really mean to recursivll y set acl on
objects of oss :// dest / a ( y or N )? y
Succeed : Total 3 objects . Setted acl on 3 objects .
0 . 963934 ( s ) elapsed
```

- Configure the meta of an object

Ossutil uses the `set-meta` command to configure the meta information of an object. You can use the `-r` option to configure the metas of objects in batches.

For more information, see `ossutil help set-meta`.

```
./ossutil set -meta oss :// dest / a x - oss - object - acl
: private - u
```

- View the object description (meta)

Ossutil uses the `stat` command to view the object description (meta).

For more information, see `ossutil help stat`.

```
$. /ossutil stat oss :// dest / a
ACL : default
Accept - Ranges : bytes
Content - Length : 230
Content - Md5 : + 5vbQC / MSQK0xXSiy KBZog ==
Content - Type : applicatio n / octet - stream
Etag : FB9BDB402F CC4902B4C5 74A2C8A059
A2
Last - Modified : 2017 - 01 - 13 15 : 14 : 22 +
0800 CST
Owner : aliyun
X - Oss - Hash - Crc64ecma : 1248880804 6134286088
```

```
X - Oss - Object - Type          : Normal
0 . 125417 ( s ) elapsed
```

- Restore an object from the frozen state to the readable state

Ossutil uses the `restore` command to restore an object from the frozen state to the readable state. You can use the `-r` option to restore objects from the frozen state to the readable state in batches.

For more information, see `ossutil help restore`.

```
$. / ossutil restore oss :// utiltest / a
0 . 037729 ( s ) elapsed
```

- Create a symbolic link

Ossutil uses the `create-symlink` command to create a symbolic link.

For more information, see `ossutil help create-symlink`.

```
$. / ossutil create - symlink oss :// utiltest / b a
0 . 037729 ( s ) elapsed
```

- Read the description of a symbolic link file

Ossutil uses the `read-symlink` command to read the description of a symbolic link file.

For more information, see `ossutil help read-symlink`.

```
$. / ossutil read - symlink oss :// utiltest / b
Etag          : D7257B62AA 6A26D66686 391037B7D6 1A
Last - Modified : 2017 - 04 - 26 15 : 34 : 27 + 0800
              CST
X - Oss - Symlink - Target : a
0 . 112494 ( s ) elapsed
```

3.7 Multipart-related commands

By using `ossutil`, you can list the IDs of unfinished multipart upload tasks (UploadID), delete files uploaded to a specified object, and delete the UploadIDs of unfinished multipart upload tasks.

For more information about multipart upload, see [Multipart upload](#).



Note:

- Before running the following commands, update your ossutil to the latest version and run the `config` command to configure the AK. For more information, see [Quick start](#).
- Ossutil automatically uses the multipart upload method but not the UploadPart command when uploading or copying large objects.

- List UploadIDs.

Run the `ls` command with the `- m` parameter to list the UploadIDs of all unfinished multipart upload tasks initiated to objects with the specified prefix.

```
$ ossutil ls oss :// bucket1 / obj1 - m
InitiatedT ime                               UploadID
      ObjectName
2017 - 01 - 13  03 : 45 : 26  + 0000  CST      15754AF798
0C4DFB8193 F190837520 BB      oss :// bucket1 / obj1
2017 - 01 - 13  03 : 43 : 13  + 0000  CST      2A1F9B4A95
E341BD9285 CC42BB950E E0      oss :// bucket1 / obj1
UploadId Number is : 2
0 . 070070 ( s ) elapsed
```

Run the `ls` command with the `- a` parameter to list the UploadIDs of all unfinished multipart upload tasks initiated to objects with the specified prefix and the uploaded objects with the prefix.

```
$ ossutil ls oss :// bucket1 / obj1 - a
LastModifi edTime                               Size ( B )   StorageCla ss
      ETAG                               ObjectName
2015 - 06 - 05  14 : 36 : 21  + 0000  CST      241561
Standard      6185CA2E8E B8510A61B3 A845EAFE41  74      oss
:// bucket1 / obj1 / test . txt
2016 - 04 - 08  14 : 50 : 47  + 0000  CST      6476984
Standard      4F16FDAE7A C404CEC8B7 27FCC67779 D6      oss
:// bucket1 / obj1 / sample . txt
Object Number is : 2
InitiatedT ime                               UploadID
      ObjectName
2017 - 01 - 13  03 : 45 : 26  + 0000  CST      15754AF798
0C4DFB8193 F190837520 BB      oss :// bucket1 / obj1
2017 - 01 - 13  03 : 43 : 13  + 0000  CST      2A1F9B4A95
E341BD9285 CC42BB950E E0      oss :// bucket1 / obj1
UploadId Number is : 2
```

```
0 . 091229 ( s ) elapsed
```

- Delete data in a specified object.

Run the `rm` command to delete the UploadIDs of all unfinished multiple upload tasks initiated to the specified object.

For example, run the `ls` command to list the UploadIDs of all unfinished multipart upload tasks initiated to objects in bucket1 and all objects uploaded to bucket1.

```
$ ossutil ls oss :// bucket1 - a
LastModifi edTime          Size ( B )  StorageCla ss
ETAG          ObjectName
2015 - 06 - 05  14 : 06 : 29 + 0000  CST          201933
Standard      7E2F4A7F1A C9D2F0996E 8332D5EA5B 41          oss
:// bucket1 / dir1 / obj11
2015 - 06 - 05  14 : 36 : 21 + 0000  CST          241561
Standard      6185CA2E8E B8510A61B3 A845EAFE41 74          oss
:// bucket1 / obj1 / test . txt
2016 - 04 - 08  14 : 50 : 47 + 0000  CST          6476984
Standard      4F16FDAE7A C404CEC8B7 27FCC67779 D6          oss
:// bucket1 / sample . txt
Object Number is : 3
InitiatedT ime          UploadID
ObjectName
2017 - 01 - 13  03 : 45 : 26 + 0000  CST          15754AF798
0C4DFB8193 F190837520 BB          oss :// bucket1 / obj1
2017 - 01 - 13  03 : 43 : 13 + 0000  CST          2A1F9B4A95
E341BD9285 CC42BB950E E0          oss :// bucket1 / obj1
2017 - 01 - 13  03 : 45 : 25 + 0000  CST          3998971ACA
F94AD9AC48 EAC1988BE8 63          oss :// bucket1 / obj2
2017 - 01 - 20  11 : 16 : 21 + 0800  CST          A20157A7B2
FEC4670626 DAE0F4C007 3C          oss :// bucket1 / tobj
UploadId Number is : 4
0 . 191289 ( s ) elapsed
```

Run the `rm` command with the `- m` parameter to delete the UploadID of the specified unfinished multipart upload task.

```
$. / ossutil rm - m oss :// bucket1 / obj1 / test . txt
Succeed : Total 1 uploadIds . Removed 1 uploadIds .
0 . 900715 ( s ) elapsed
```

Run the `rm` command with the `- m` and `- r` parameters to delete the UploadIDs of all unfinished multipart upload tasks initiated to objects with the specified prefix.

```
$. / ossutil rm - m oss :// bucket1 / ob - r
Do you really mean to remove recursive y multipart
uploadIds of oss : bucket1 / ob ( y or N )? y
Succeed : Total 4 uploadIds . Removed 4 uploadIds .
```

```
1 . 922915 ( s ) elapsed
```

Run the `rm` command with the `-a` and `-r` parameters to delete the UploadIDs of all unfinished multipart upload tasks initiated to objects with the specified prefix and all uploaded objects with the specified prefix.

```
$. /ossutil rm oss :// hello - hangzws - 1 / obj - a - r  
Do you really mean to remove recursively objects  
and multipart uploadIds of oss :// obj ( y or N )? y  
Succeed : Total 1 objects , 3 uploadIds . Removed 1  
objects , 3 uploadIds .
```

4 ossimport

4.1 Architecture and configuration

Overview

The OssImport tool allows you to migrate data stored locally or in other cloud storage systems to the OSS. It has the following features:

- Supports a rich variety of data sources including local drives, Qiniu, Baidu BOS, AWS S3, Azure Blob, Youpai Cloud, Tencent Cloud COS, Kingsoft KS3, HTTP, and OSS, and can be expanded as needed.
- Supports resumable data transfers.
- Supports throttling.
- Supports migrating objects after a specified time point or with a specified prefix.
- Supports parallel data uploads and downloads.
- Supports standalone and distributed modes. The standalone mode is easy to deploy and use, and the distributed mode is suitable for large-scale data migration.

Environment

- Java 1.7 and later

Architecture

The OssImport has two deployment modes available: standalone mode and distributed mode.

- The standalone mode is sufficient for small-scale data migration which smaller than 30 TB. [Download](#)
- Distributed mode is recommended for larger data migrations. [Download](#)
- Standalone

The master, worker, tracker, and console run on the same machine. There is only one worker in the system. We have encapsulated and optimized the deployment and execution of the standalone mode and the standalone deployment and

execution are both easy. In standalone mode, the master, worker, tasktracker, and console modules are packaged into `ossimport2.jar`.

The file structure in standalone mode is as follows:

```

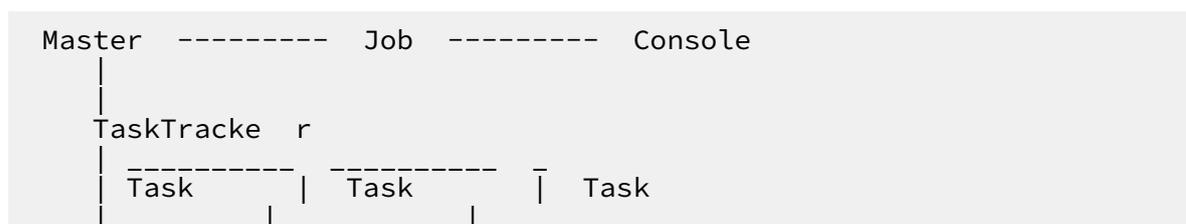
ossimport
├── bin
│   └── ossimport2.jar # The JAR including master,
│                       worker, tracker, and console modules
├── conf
│   └── local_job.cfg # Standalone job configuration
├── file
│   └── sys.properties # Configuration file for the
│                       system running
├── console.bat # Windows command line, which
│               can run distributed call-in tasks
├── console.sh # Linux command line, which can
│              run distributed call-in tasks
├── import.bat # The configuration file for
│              one-click import and execution in Windows is the
│              data migration job configured in conf/local_job.cfg
│              , including start, migration, validation, and retry
├── import.sh # The configuration file of
│              one-click import and execution in Linux is the
│              data migration job configured in conf/local_job.cfg
│              , including start, migration, validation, and retry
├── logs # Log directory
└── README.md # Description documentat ion . We
               recommend that you carefully read the documentat ion
               before using this feature

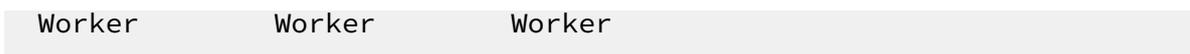
```

Note:

- The `import.bat` or `import.sh` file is a one-click import script and can be run directly after you complete modification to `local_job.cfg`.
 - The `console.bat` or `console.sh` is the command line tool and can be used for distributed execution of commands.
 - Run scripts or commands in the `ossimport` directory, that is, the directory at the same level as the `*.bat` / `*.sh` file.
- Distributed

The OssImport is based on the master-worker distributed architecture, as shown in the following figure:





In the figure:

- **Job:** The data migration jobs submitted by users. For users, one job corresponds to one configuration file `job . cfg`.
- **Task:** A job can be divided into multiple tasks by data size and number of files . Each task migrates a portion of files. The minimal unit for dividing a job into tasks is a file. One file cannot be split into multiple tasks.

The OssImport tool modules are listed in the following table:

Role	Description
Master	<p>The master is responsible for splitting a job into multiple tasks by data size and number of files. The data size and number of files can be configured in <code>sys.properties</code>. The detailed process for splitting a job into multiple tasks is as follows:</p> <ol style="list-style-type: none"> 1. The master node scans the full list of files to be migrated from the local/ other cloud storage devices. 2. The master splits the full file list into tasks by data size and the number of files and each task is responsible for the migration or validation for a part of files.
Worker	<ul style="list-style-type: none"> - The worker is responsible for file migration and data validation of tasks. It pulls the specific file from the data source and uploads the file to the specified directory to the OSS. You can specify the data source to be migrated and the OSS configuration in <code>job.cfg</code> or <code>local_job.cfg</code>. - Worker data migration supports limiting traffic and specifying the number of concurrent tasks. You can configure the settings in <code>sys.properties</code>.

Role	Description
TaskTracker	TaskTracker is abbreviated to Tracker. It is responsible for distributing tasks and tracking task statuses.
Console	The console is responsible for interacting with users and receiving command display results. It supports system management commands such as deploy, start, and stop, and job management commands such as submit, retry, and clean.

In distributed mode, you can start multiple worker nodes for data migration. Tasks are evenly allocated to the worker nodes and one worker node can run multiple tasks. One machine can only start one worker node. The master is started at the same time as the first worker node configured in `workers`, and the tasktracker and console also run on the machine.

The file structure in distributed mode is as follows:

```

ossimport
├── Miss -- Bin
│   ├── console . jar      # The JAR package of the
│   │   console module
│   ├── master . jar      # The JAR package of the master
│   │   module
│   ├── tracker . jar     # The JAR package of the
│   │   tracker module
│   └── worker . jar      # The JAR package of the worker
│       module
├── conf
│   ├── job . cfg         # The template of the job
│   │   configurat ion file
│   ├── sys . properties # Configurat ion file for the
│   │   system running parameters
│   └── workers          # Worker list
├── console . sh         # The command line tool .
│   Currently it only supports Linux
├── logs                # Log directory
└── README . md         # Descriptio n documentat ion . We
    recommend that you carefully read the documentat ion
    before using the feature

```

Note:

- The distributed command line tool `console.sh` currently only supports Linux and does not support Windows.

Configuration files

In standalone mode, two configuration files are used: `sys.properties` and `local_job.cfg`. In distributed mode, three configuration files are used: `sys.properties`, `local_job.cfg`, and `workers`. Specifically, `local_job.cfg` and `job.cfg` are identical, except in name. The `workers` file is exclusive to the distributed environment.

- `sys.properties`

System running parameters.

Field	Meaning	Description
<code>workingDir</code>	Working directory.	<ul style="list-style-type: none"> - The directory after the tool kit is extracted. - Do not modify this option in standalone mode. - The working directory for each machine in distributed mode must be the same.
<code>workerUser</code>	The worker machine SSH user name.	<ul style="list-style-type: none"> - If you have configured <code>privateKeyFile</code>, the <code>privateKeyFile</code> is used in priority. - If <code>privateKeyFile</code> is not configured, the <code>workerUser/workerPassword</code> combination is used. - Do not modify this option in standalone mode.
<code>workerPassword</code>	The worker machine SSH user password.	Do not modify this option in standalone mode.

Field	Meaning	Description
privateKeyFile	The file path of the private key.	<ul style="list-style-type: none">- If you have established an SSH channel, you can specify the public key file path. Otherwise, leave it empty.- If you have configured privateKeyFile, the privateKeyFile is given priority.- If privateKeyFile is not configured, the workerUser/workerPassword is used.- Do not modify this option in the standalone mode.
sshPort	The SSH port.	The default value is 22. It does not usually need to be changed. Do not modify this option in standalone mode.

Field	Meaning	Description
workerTaskThreadNum	The maximum number of threads for the worker to run tasks.	<ul style="list-style-type: none"> - This parameter is related to the machine memory and network. Recommended value is 60. - The value can be increased, for example to 150 for physical machines. If the network bandwidth is already full, do not increase the value further. - If the network is poor, lower the value as appropriate to, for example, 30. This way, you can avoid the time-out of a large number of requests from request competition.
workerMaxThroughput(KB/s)	The data migration traffic ceiling on the worker node.	This value limits the traffic. The default value 0 indicates that no traffic limitations are imposed.
dispatcherThreadNum	The number of threads for task distribution and status confirmation of the tracker.	The default value must be enough. You don't need to change the default value if you have no special requirements.
workerAbortWhenUncatchedException	Whether to skip or cancel in case of an unknown error.	Unknown errors are skipped by default.
workerRecordMd5	Whether to use metadata x-oss-meta-md5 to log the MD5 value of the migrated file in the OSS. The default setting is no.	It is mainly used for file data validation using MD5.

- job.cfg

Data migration job configuration. The local_job.cfg and job.cfg options are identical except in name.

Field	Meaning	Description
jobName	The job name, a string.	<ul style="list-style-type: none"> - The unique identifier of the job. The naming rule is [a-zA-Z0-9_-]{4,128}. It supports the submission of multiple jobs of different names. - If you submit a job with the same name as another job, the system prompts that the job already exists. You are not allowed to submit a job of the same name before you clean the original job with the name.
jobType	The job type, a string.	<p>There are two types: import and audit. The default value is import.</p> <ul style="list-style-type: none"> - import: Run the data migration and validate the migrated data for consistency. - audit: Only validate data consistency.
isIncremental	Whether to enable incremental migration mode, a Boolean value.	<ul style="list-style-type: none"> - Default value: False. - If it is set to true, incremental data is rescanned at the interval specified by incrementalModeInterval (unit: second) and synchronized to the OSS.

Field	Meaning	Description
incrementalModeInterval	Synchronization interval in incremental mode, an integer value. Unit: second.	Valid when isIncremental=true. The minimum configurable interval is 900 seconds. We do not recommend you configure it to a value smaller than 3,600 seconds as that wastes a large number of requests and lead to additional system overhead.
importSince	Migrate data later than this time value, an integer value. Unit: second.	<ul style="list-style-type: none"> - This time value is a Unix timestamp, that is, the number of seconds since UTC 00:00 on January 1, 1970. You can get the value through the <code>date +%s</code> command. - The default value is 0, indicating to migrate all the data.

Field	Meaning	Description
srcType	The synchronization source type, a string. Case sensitive.	<p>Currently this parameter supports 10 types including local, oss, qiniu, bos, ks3, s3, youpai, HTTP, cos, and azure.</p> <ul style="list-style-type: none"> - local: Migrate data from a local file to the OSS. You only need to enter the srcPrefix for this option and do not need to enter srcAccessKey, srcSecretKey, srcDomain, and srcBucket. - Migrate data from one bucket to another. - qiniu: Migrate data from Qiniu cloud storage to the OSS. - bos: Migrate data from Baidu cloud storage to the OSS. - ks3: Migrate data from Kingsoft cloud storage to the OSS. - s3: Migrate data from AWS S3 to the OSS. - youpai: Migrate data from Youpai Cloud to the OSS. - HTTP: Migrate data to the OSS through the provided HTTP link list. - cos: Migrate data from the Tencent cloud storage COS to the OSS. - azure: Migrate data from Azure Blob to the OSS.

Field	Meaning	Description
srcAccessKey	The source AccessKey, a string.	Enter the AccessKey of the data source if srcType is set to oss, qiniu, baidu, ks3, or s3. <ul style="list-style-type: none">- or the local and HTTP types, this option can be left empty.- For youpai and azure types, enter the AccountName.
srcSecretKey	The source SecretKey, a string.	Enter the SecretKey of the data source if srcType is set to oss, qiniu, baidu, ks3, or s3. <ul style="list-style-type: none">- For the local and HTTP types, this option can be left empty.- youpai: Enter the operator password.- azure: Enter the AccountKey.

Field	Meaning	Description
srcDomain	Source endpoint.	<p>This configuration item is not required if the srcType is set to local or HTTP.</p> <ul style="list-style-type: none"> - oss: The domain name obtained from the console. It is a second-level domain name without the bucket prefix. A full list can be found at domain name list. - qiniu: The domain name of the corresponding bucket obtained from the Qiniu console. - bos: The Baidu BOS domain name, such as <code>http://bj.bcebos.com</code> or <code>http://gz.bcebos.com</code>. - ks3: Kingsoft KS3 domain name, such as <code>http://kss.ksyun.com</code>, <code>http://ks3-cn-beijing.ksyun.com</code> or <code>http://ks3-us-west-1.ksyun.com</code>. - The S3 and AWS S3 domain names of various regions can be found at S3 Endpoint. - youpai: The domain name of the Youpai Cloud, such as automatic identification of the optimal channel of
154		<p><code>http://www.20app.com</code>, or telecommunication</p>

Field	Meaning	Description
srcBucket	The name of the source bucket or the container.	This configuration item is not required if the srcType is set to local or HTTP. azure: Enter the container name in Azure Blob, and enter the bucket name for others.
srcPrefix	The source prefix, a string . The default value is empty.	If the srcType is set to local, enter the local directory in full, separated, and ended by /, such as c:/example/ or /data/example/. If the srcType is oss, qiniu, bos , ks3, youpai, or s3, the value is the prefix of the object to be synchronized , without the bucket name , such as data/to/oss/. If you want to synchronize all the objects, leave the srcPrefix empty.
destAccessKey	The destination AccessKey, a string.	To view the OSS AccessKeyID, log on to the console . .
destSecretKey	The destination SecretKey , a string.	To view the OSS AccessKeySecret, log on to the console . .

Field	Meaning	Description
destDomain	Destination endpoint, a string.	Obtained from the console . It is a second-level domain name without the bucket prefix. A full list can be found at domain name list . .
destBucket	The destination bucket, a string.	The OSS bucket name. It does not need to end with /.

Field	Meaning	Description
destPrefix	The destination prefix, a string. The default value is empty.	<ul style="list-style-type: none"> - The destination prefix . The default value is empty in which case the objects are placed in the destination bucket. - If you want to synchronize data to a specific directory on the OSS, end the prefix with /, such as data/in/oss/. - Note that the OSS does not support / as the object prefix, so do not set destPrefix to start with /. - A local file in the path srcPrefix+relativePath is migrated to destDomain/destBucket/destPrefix + relativePath on the OSS. - An object on the cloud in the path srcDomain/srcBucket/srcPrefix+relativePath is migrated to destDomain/destBucket/destPrefix + relativePath on the OSS.

Field	Meaning	Description
taskObjectCountLimit	The maximum number of files in a task, an integer. The default value is 10,000.	This configuration option affects the concurrency of the executed jobs. Generally the configuration is set to the total number of files/total number of workers/number of migration threads (workerTaskThreadNum) and the maximum number is 50,000. If the total number of files is unknown, use the default value.
taskObjectSizeLimit	The maximum data size in a task, an integer. Unit: bytes. The default value is 1 GB.	This configuration option affects the concurrency of the executed jobs. Generally the configuration is set to the total data size/total number of workers/number of migration threads (workerTaskThreadNum). If the total data size is unknown, use the default value.
isSkipExistFile	Whether to skip the existing objects during data migration, a Boolean value.	If it is set to true, the objects are skipped according to the size and LastModifiedTime. If it is set to false, the existing objects are overwritten. The default value is false. This option is invalid when jobType is set to audit.

Field	Meaning	Description
scanThreadCount	The number of threads for parallel file scanning, an integer. The default value is 1.	This configuration option is related to file scanning efficiency. Do not modify the configuration if you have no special requirements.
maxMultiThreadScanDepth	The maximum allowable depth of directories for the parallel scan, an integer. The default value is 1.	<ul style="list-style-type: none"> - The default value of 1 indicates parallel scan on top-level directories. - Do not modify this configuration if you have no special requirements. If the value is configured too large, the job may fail to run normally.
appId	The appId of the Tencent COS, an integer.	Valid when srcType is set to cos.

Field	Meaning	Description
httpListFilePath	The absolute path of the HTTP list file, a string.	<ul style="list-style-type: none"> - Valid when srcType is set to HTTP. When the source is an HTTP link address, you are required to provide the absolute path of the file with the HTTP link address as the content , such as c:/example/http.list. - The HTTP link in the file must be divided into two columns separated by spaces, representing the prefix and the relative path on the OSS after the upload respectively, such as c:/example/http.list which contains the following content: <pre style="margin: 0; padding-left: 20px;">http :// mingdi - hz . oss - cn - hangzhou . aliyuncs . com / aa / bb . jpg http :// mingdi - hz . oss - cn - hangzhou . aliyuncs . com / cc / dd . jpg .</pre> The object names for the two rows after they are migrated to the OSS are destPrefix + bb.jpg and destPrefix + cc/dd.jpg respectively.

• Workers

The workers is exclusive to the distributed mode and every IP address is a row, such as:

```
192 . 168 . 1 . 6
```

```
192 . 168 . 1 . 7
192 . 168 . 1 . 8
```

Note:

- In the preceding configuration, the `192 . 168 . 1 . 6` in the first line must be master, that is, the master, worker, and TaskTracker are started on `192 . 168 . 1 . 6` and the console also needs to be executed on the machine.
- Make sure that the user name, logon mode, and working directory of multiple worker modes are the same.

Configuration file example

The data migration task profile for a distributed deployment is shown in the following table, and the configuration file name for a stand-alone machine is `local_job . cfg`, there is no difference between a configuration item and a distributed deployment.

Migration type	Configuration File	Description:
Migrate locally to OSS	job.cfg	Srcprefix is an absolute path at the end of/, such <code>D :/ work / oss / data /, / home / user / work / oss / data /</code>
Migrating from seven bull cloud storage to OSS	job.cfg	Srcprefix and DESTIN prefix can be configured to be empty; if not empty, end with / such as <code>destPrefix = docs /</code>
Transfer from Baidu Bos to OSS	job.cfg	Srcprefix and DESTIN prefix can be configured to be empty; if not empty, end with /, such as <code>destPrefix = docs /</code>
Migrating from AWS S3 to OSS	job.cfg	domain names for S3
Move from cloud storage to OSS again	job.cfg	Srcaccesskey/Scanner enters the operator account number and password

Migration type	Configuration File	Description:
Migrating from Tencent COs to OSS	job.cfg	Srcdomain please follow V4 version, such as <code>srcDomain = sh .</code> Srcprefix can be empty, when not empty, start and end with /, such as <code>srcPrefix =/ docs /</code>
Migrating from azure blob to OSS	job.cfg	Srcaccesskey/srcsecretkey fill storage cun chu and key; srcdomain enters connection string Endpointsuffix, such as <code>core . chinacloud api . cn .</code>
Migrating from OSS to OSS	job.cfg	It is suitable for data migration between different regions, between Different Storage types , and between different prefixes; it is recommended to deploy on ECS and use domain names with internal to save on traffic.

4.2 Standalone deployment

Standalone deployment supports Linux and Windows.

Download

Download the tool for standalone deployment: [ossimport-2.3.4.zip](#). Download the tool to a local directory and use a tool or run the `unzip` command to unzip the files. The file structure after unzipping is as follows:

```

ossimport
├── bin
│   ├── ossimport2 . jar # The JAR including master ,
│   │   worker , tracker , and console modules
│   └── conf
│       ├── local_job . cfg # The job configuration file
│       └── sys . properties # Configuration file of the
│           system running parameters
└── console . bat # Windows command line , which can
    run distributed call - in tasks

```

```

├─ console . sh          # Linux command line , which can
run distribute d call - in tasks
├─ import . bat         # The configurat ion file for
one - click import and execution in Windows is the
data migration job configured in conf / local_job . cfg ,
including start , migration , validation , and retry
├─ import . sh         # The configurat ion file for
one - click import and execution in Linux is the
data migration job configured in conf / local_job . cfg ,
including start , migration , validation , and retry
├─ logs                # Log directory
└─ README . md         # Descriptio n documentat ion . We
recommend that you carefully read the documentat ion
before using the feature

```

Configuration

The standalone version has two configuration files: `conf / sys . properties` and `conf / local_job . cfg` .

- Do not change the configuration items in `conf / sys . properties` :
`workingDir` , `workerUser Name` , `workerPass word` , and `privateKey File` .
- Do not change the name and location of `conf / local_job . cfg` and the `jobName` configuration item in it.

Configure other items appropriately.



Note:

Confirm the parameters in `sys . properties` and `local_job . cfg` before submitting the job. The parameters in the job are not allowed to be changed after the job is submitted.

Running

In standalone mode, a data migration job has two execution modes: one-click import and step-by-step execution.

One-click import encapsulates all the steps and data migration can then be completed following the prompts of the script.



Note:

We recommend you use one-click import if you use ossimport for the first time.

Step-by-step execution includes executing the starting service, submitting the job and retrying failed tasks.

- One-click import

1. To run one-click import, run `import . bat` in cmd.exe in Windows, and run `bash import . sh` in Linux.
2. If you previously run this job, you are asked if you want to continue the job from the last breakpoint or if you want to run a new synchronization job. If you initiate a new data migration job, or have modified the synchronized source end/destination end, run the synchronization job again.
3. After a job starts in Windows, a new cmd window appears showing the synchronization job in progress and the log. The job status in the old window is refreshed every 10 seconds. Do not close these two windows during the data migration process. In Linux, the preceding process is run in the background.
4. When the job is complete, if a task failed, you are asked if you want to retry. Enter `y` to retry or `n` to skip this step and exit.
5. To see why the upload failed, open the file `master / jobs / local_test / failed_tasks /< tasktaskid >/ audit . log` and check the cause of the failure.

- Step-by-step execution

1. Clear jobs with the same name. If you have run job with the same name before and want to run the job again, first clear the job with the same name. If you have never run the job or you want to retry a failed job, do not run the clear command. In Windows, run `console . bat clean` in cmd.exe. In Linux, run `bash console . sh clean`.
2. Submit the data migration job. OssImport does not support submitting jobs of the same name. If jobs with the same name exist, clear the job with the same name first. The configuration file for the submitted job is `conf / local_job . cfg`, and the default job name is `local_test`. To submit a job, run `console`

- `. bat submit` in cmd.exe in Windows, and run `bash console . sh submit` in Linux.
3. Start the service. Run `console . bat start` in cmd.exe in Windows, and run `bash console . sh start` in Linux.
 4. View the job status. Run `console . bat start` in cmd.exe in Windows, and run `bash console . sh start` in Linux.
 5. Retry a failed task. Tasks may fail due to network issues or other causes. Only failed tasks are retried. Run `console . bat retry` in cmd.exe in Windows, and run `bash console . sh retry .`
 6. Stop the service. Close the `%JAVA_HOME%/bin/java.exe` window in Windows, and run `bash console . sh stop` in Linux.

**Note:**

We recommend that you use one-click import for data migration if you have no special requirements.

- Common causes of failure
 - A file in the source directory was modified during the upload process. This cause is indicated by a `SIZE_NOT_M ATCH` error in `log / audit . log` . In this case, the old file has been uploaded successfully, but the changes have not been synchronized to the OSS.
 - A source file was deleted during the upload process, leading to download failure .
 - A source file name does not conform to naming rules of the OSS (file name cannot start with / or be empty), leading to upload failure.
 - The data source file failed to be downloaded.
 - The program exited unexpectedly and the job status is Abort. If this happens, contact after-sales technical support.
- Job statuses 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 statuses. After a job is completed, the `ossimport` directory contains the following:

```
ossimport
├── bin
│   └── ossimport2 . jar      # The standalone version JAR
```

```

├── conf
│   ├── local_job . cfg      # The job configuration file
│   └── sys . properties    # Configuration file of the
system running parameters
├── console . sh           # The command line tool
├── import . sh           # One - click import script
├── logs
│   ├── import . log       # Migration logs
│   ├── job_stat . log     # Job status record
│   └── ossimport2 . log   # Running log of the
standalone version
├── submit . log          # Job submission record
├── master
│   ├── jobqueue           # Store jobs that have
not been fully split
│   └── jobs               # Store the job running
status
│   ├── local_test        # Job name
│   │   ├── checkpoint s  # The checkpoint record
│   │   │   └── 0
│   │   │       └── 034DC9DD28 60B0CFE884 242BC6FF92 E7 .
cpt
│   │   ├── dispatched   # Tasks that have been
assigned to the workers but haven ' t been fully
run
│   │   └── localhost
│   │       ├── failed_tas ks # Tasks that failed to
run
│   │       ├── pending_ta sks # Tasks that have not
been assigned
│   │       └── succeed_ta sks # Tasks that run
successfully
│   │           ├── A41506C07B F1DF2A3EDB 4CE31756B9 3F_1499744
514501 @ localhost
│   │           ├── audit . log # The task running
log . You can view the error causes in the log
│   │           └── DONE      # Mark of successful
tasks
│   │               ├── error . list # The task error list
│   │               └── STATUS # The task status
marker file . The content is Failed or Completed
│   │                   ├── TASK # The task descriptio n
information
├── worker # Status of the task being run by
the worker . After running , tasks are managed by the
master
│   ├── jobs
│   │   └── local_test
│   │       └── tasks

```

**Note:**

- For job running information, view `logs / ossimport2 . log` or `logs / import . log` .
- For the task failure cause, view `master / jobs / ${ JobName } / failed_tas ks / ${ TaskName } / audit . log` .

- For failed task files, view `master / jobs /${ JobName }/ failed_tasks / ${ TaskName }/ error . list .`
- The preceding log files are for reference only. Do not deploy your services and applications entirely based on them.

FAQ

See [FAQ](#).

4.3 Distributed deployment

Download

Distributed deployment currently only supports Linux, and does not support Windows.

Download the tool for distributed deployment: [ossimport-2.3.4.tar.gz](#).

Download the tool to a local directory and use the command `tar -zxvf ossimport - 2 . 3 . 4 . tar . gz - C $ HOME / ossimport` to unzip the files. The file structure after the unzipping is as follows:

```

ossimport
├── bin
│   ├── console . jar      # The JAR package of the
│   └── module
│       ├── master . jar  # The JAR package of the master
│       └── module
│           ├── tracker . jar  # The JAR package of the
│           └── worker . jar  # The JAR package of the worker
├── conf
│   ├── job . cfg         # The template of the job
│   └── sys . properties  # Configurat ion file of the
│       system running parameters
├── console . sh         # Worker list
├── logs                 # The command line tool .
├── README . md         # Log directory
└── workers              # Descriptio n documentat ion . Read
    it carefully before use

```

Note:

- `OSS_IMPORT_HOME`: The root directory of ossImport. By default the directory is the `$HOME/ossimport` in the unzip command. You can also run the `export`

`OSS_IMPORT_HOME =< dir >` command or modify the system configuration file `$ HOME /. bashrc` to set the directory.

- `OSS_IMPORT_WORK_DIR`: The ossImport working directory. You can specify the directory through the configuration item `workingDir` in `conf / sys . properties`. The recommended values is `$ HOME / ossimport / workdir`.
- Use absolute paths for `OSS_IMPORT_HOME` or `OSS_IMPORT_WORK_DIR`, such as `/home /< user >/ ossimport` or `/home /< user >/ ossimport / workdir`.

Configuration

The distributed version has three configuration files: `conf / sys . properties`, `conf / job . cfg`, and `conf / workers`. For descriptions of the configuration items, see the Introduction chapter.

- `conf / job . cfg`: The configuration file template for the job in distributed mode. Modify the values according to the actual parameters before data migration.
- `conf / sys . properties`: The configuration file for the system run parameters, such as the working directory and the worker running parameters.
- `conf / workers`: The worker list.



Note:

- Confirm the parameters in `sys . properties` and `job . cfg` before submitting the job. The parameters in the job are not allowed to be changed after the job is submitted.
- Determine the worker list `workers` before starting the service. After the service is started, workers are not allowed to be added or deleted.

Running

- Run commands.

In distributed deployment, the general steps for job execution are as follows:

- Modify the job configuration file.
- Deploy the service.

Run `bash console . sh deploy` in Linux.



Note:

Make sure the configuration files `Conf / job . cfg` and `CONF / workers` have been modified before deployment.

- Clear jobs of the same name.

If you ran a job of the same name before and want to run the job again, clear the job with the same name first. If you have never run the job or you want to retry the tasks of a failed job, do not run the clear command. Run `bash console . sh clean job_name` in Linux.

- Submit the data migration job.

OssImport does not support submitting jobs of the same name. If jobs with the same name exist, use the `clean` command to clean the job with the same name first. To submit a job, you must specify the job configuration file. The job's configuration file template is `conf / job . cfg`. We recommend that you modify the settings based on the template. Run `bash console . sh submit [job_cfg_file]` in Linux and submit the job with the configuration file `job_cfg_file`. The `job_cfg_file` is an optional parameter. If not specified, the parameter is `$ OSS_IMPORT _HOME / conf / job . cfg` by default. The `$ OSS_IMPORT _HOME` is by default the directory where the `console . sh` file is located.

- Start the migration service.

Run `bash console . sh start` in Linux.

- View the job state.

Run `bash console . sh stat` in Linux.

- Retry failed tasks.

Tasks may fail to run because of network issues or other causes. Only failed tasks are retried. Run `bash console . sh retry [job_name]` in Linux. The `job_name` parameter is optional. If it is specified, tasks of failed jobs are retried. If it is not specified, tasks of all jobs are retried.

- Stop the migration job.

Run `bash console . sh stop` in Linux.



Note:

- When the `bash console . sh` parameter has an error, `console . sh` automatically prompts the command format.
- We recommend that you use absolute paths for directories of the configuration file and submitted jobs.
- The configuration for jobs (that is, the configuration items in `job . cfg`) cannot be modified after submitted.

- Common causes of job failure

- A file in the source directory was modified during the upload process. This cause is indicated by a `SIZE_NOT_MATCH` error in `log / audit . log`. In this case, the old file has been uploaded successfully, but the changes have not been synchronized to the OSS.
- A source file was deleted during the upload process, leading to the download failure.
- A source file name does not conform to naming rules of the OSS (file name cannot start with / or be empty), leading to the upload failure to the OSS.
- The data source file fails to be downloaded.
- The program exits unexpectedly and the job state is Abort. If this happens, contact after-sales technical support.

- Job states 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 states. After a job is completed, the `workdir` directory contains the following:

```

workdir
├── bin
│   ├── console . jar      # The JAR package of the
│   │   console          module
│   ├── master . jar      # The JAR package of the
│   │   master          module
│   ├── tracker . jar     # The JAR package of the
│   │   tracker        module
│   └── worker . jar      # The JAR package of the
│       worker        module
├── conf
│   ├── job . cfg         # The template of the job
│   │   configurat ion  file
│   ├── sys . properties # Configurat ion file of the
│   │   system          running parameters
│   └── workers          # Worker list
├── logs
│   ├── import . log     # Migration logs
│   ├── master . log     # Master logs
│   └── tracker . log    # Tracker logs

```

```

├── worker . log      # Worker logs
├── master
│   ├── jobqueue     # Store jobs that have
│   │   not been fully split
│   ├── jobs         # Store the job running
│   │   state
│   │   ├── xctooss  # Job name
│   │   │   ├── checkpoint s # The checkpoint record
│   │   │   │   ├── master splits the job to tasks
│   │   │   │   │   ├── 0
│   │   │   │   │   └── ED09636A6E A24A292460 866AFDD7A8 9A .
│   │   │   cpt
│   │   ├── dispatched # Tasks that have been
│   │   │   assigned to the workers but haven ' t been fully
│   │   │   run
│   │   └── 192 . 168 . 1 . 6
│   │       ├── failed_tas ks # Tasks that failed to
│   │       │   run
│   │       └── A41506C07B F1DF2A3EDB 4CE31756B9 3F_1499348
│   │           973217 @ 192 . 168 . 1 . 6
│   │               ├── audit . log # The task running
│   │               │   log . You can view the error causes in the log
│   │               │   ├── DONE # Mark of successful
│   │               │   │   tasks . If the task fails , the mark is empty
│   │               │   ├── error . list # The task error
│   │               │   │   list . You can view the error file list
│   │               │   ├── STATUS # The task state
│   │               │   │   mark file . The content is Failed or Completed ,
│   │               │   │   indicating that the task failed or succeeded
│   │               │   └── TASK # The task descriptio
│   │               │   n informatio n
│   │               ├── pending_ta sks # Tasks that have not
│   │               │   been assigned
│   │               └── succeed_ta sks # Tasks that run
│   │                   successful ly
│   │                   ├── A41506C07B F1DF2A3EDB 4CE31756B9 3F_1499668
│   │                   │   462358 @ 192 . 168 . 1 . 6
│   │                   │   ├── audit . log # The task running
│   │                   │   │   log . You can view the error causes in the log
│   │                   │   │   ├── DONE # Mark of successful
│   │                   │   │   │   tasks
│   │                   │   │   ├── error . list # Task error list .
│   │                   │   │   │   If the task is successful , the list is empty
│   │                   │   │   ├── STATUS # The task state
│   │                   │   │   │   mark file . The content is Failed or Completed ,
│   │                   │   │   │   indicating that the task failed or succeeded
│   │                   │   │   └── TASK # The task descriptio n
│   │                   │   informatio n
│   │                   └── worker # state of the task being run by the
│   │                       worker . After running , tasks are managed by the
│   │                       master
│   │                           ├── jobs
│   │                           │   ├── local_test 2
│   │                           │   │   ├── tasks
│   │                           │   └── local_test _4
│   │                           │       ├── tasks

```

**Note:**

- For job running information, view `logs / import . log .`

- For the task failure cause, view `master / jobs /${ JobName }/ failed_tasks /${ TaskName }/ audit . log .`
- For failed task files, view `master / jobs /${ JobName }/ failed_tasks /${ TaskName }/ error . list .`
- The preceding log files are for reference only. Do not deploy your services and application entirely based on them.

FAQ

See [FAQ](#).

4.4 FAQ

- 1. `UnsupportedClassVersionError`

Exception Executing command:

```
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
r . java : 306 )
    at java . lang . ClassLoader . loadClass ( ClassLoader
r . java : 247 )
    at com . simontuffs . onejar . Boot . run ( Boot . java :
300 )
    at com . simontuffs . onejar . Boot . main ( Boot . java
: 159 )
```

Cause: the Java version is too low to be updated to 1.7 or later.

- 2. `InvocationTargetException`

Submit task reporting exceptions using the `submit` command:

```
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 . java : 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.NullPointe rException
    at com.aliyun.ossimport2.config.JobConfig.load
( JobConfig . java : 44 )
    at com.aliyun.ossimport2.OSSImport2.doSubmitJo
b ( OSSImport2 . java : 289 )
    at com.aliyun.ossimport2.OSSImport2.main (
OSSImport2 . java : 120 )
... 6 more

```

Reason: Check to see if the items in the configuration file are deleted or commented out, please enter items that do not need to be configured after the equal sign and do not need to be deleted.

- 3. too many open files

Reason: `ulimit -n` view system handle.

- If the value is less than 10 thousand, you can restart the process through `ulimit -n 65536 ;`
- If it was already set up relatively large, then use `sudo losf -n` to troubleshoot which processes have opened the handle.

- 4 Windows return seconds after Windows starts

Cause: Most cases are caused by Java not installed or version less than 1.7, or by configuration file errors.

- 5. No jobs is running or finished

When the `submit` command completes the task, use `stat`. View task status always displays:

```

bash console . sh stat
[ WARN ] List files dir not exist : / home /< user >/
ossimport / workdir / master / jobs /
no jobs is running or finished .

```

Reason:

- The job was just submitted, and the master needs to scan the list of files first, when the task is not actually generated and distributed, printing the log is normal;
- After a long period of time, the error is still printed, usually without `start` . Command to start the process or to exit unexpectedly after the process has

started. If you do not start the service, you only need to use `start`; otherwise, take a look `logs / ossimport . log`, find the cause of the exception and resolve it before you start the service process.

- 6. The `STAT` command always displays `scanfinished: false`

Observe whether the total number of tasks is increasing:

- If there is more in the process, it is that the file list of the job is not complete, there are also new files in the list;
- Always unchanged, `scanfinished` will never be true if the job is configured with incremental Mode To scan the list of files regularly, depending on the interval configured by the user, check for new or modified documents;
- If it is not an incremental mode, the number of tasks does not increase, and the log is checked for exceptions.

- 7. The service process was dropped, but the log did not output the exception

Reason: if the machine's available memory is less than 2 GB, the big probability is that there's not enough memory to be killed. Check the `dmsg`. Log whether there is a record of insufficient memory to be killed.

- 8. What needs to be done to restart the service after the process has been hung or killed?

Call `start directly` The command starts the service, and the job that has been submitted does not need to be resubmitted, as long as it does not call the `clean` command, all submitted jobs have breakpoint records that do not redo the work that has been done.

- 9. Complete the task the OSS console displays a smaller amount of data than the source

There is no change in the size of the bucket in the OSS console after the job has all been successfully uploaded or used locally. The size of `du` statistics varies greatly. Cause: the amount of Bucket data in the OSS console is delayed for 1 hour to update. `du` The command counts the block size, which is larger than the actual file, you can count the true size of the local directory by referring to the following command: `ls -lR < directory absolute path > | grep "\- rw "`
`| awk '{ sum += $ 5 } END { print sum }'.`

- 10. How do I handle the failed tasks shown by stat?

Generally, you can use the `retry` command to try again.

- 11. After some failed tasks, repeated retry won't succeed.

Reason: view the file `$ work_dir / master / jobs /$ jobName / failed_tasks /$ taskName / error . list` Get the relative path of the failed file, check if the file has permission to access, whether it is deleted, is flexible, whether garbled file name, etc.

- 12. How do I upload a file with a bad file name to OSS?

Need to first use `export LANG = "< your file name encode >", ls use encode>`, `ls` after checking the file name. Command to clear the original job and resubmit the job again with the `submit` command.

- 13. `java.nio.file.AccessDeniedException`

Exception reported: `ava.nio.file.AccessDeniedException`. Cause: There is no permission to access the configuration file directory.

- 14. Task status displays 0, but job display completes

The task status displays 0, but the job display completes as follows:

```
[ 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
```

Reason:

- The `srcPrefix` fills in the error, resulting in the `List` not coming out of the file;
 - There are only directories and no files under `srcPrefix`, because the concept of directories is simulated by OSS, will not be truly uploaded.
- 15. The bucket you are attempting to access must be addressed using the specified endpoint

Log reporting exception:

```
Exception : com . aliyun . oss . OSSExcepti on : The bucket
you are attempting to access must be addressed
```

```

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 > 56EA98DE81 5804 ** 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 >

```

Reason: `srcDomain` of Bucket Or `destDomain` fill in the error, please follow the list of domain names Fill in the correct domain name.

- 16. The request signature we calculated does not match the signature you provided

Log reporting exception:

```

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 ]: xxxxxxxx
[ HostId ]: xxx . oss - cn - shanghai . aliyuncs . com

```

Reason: Check whether the `destAccess Key` , `destSecret Key` and the scanner are wrong. Please refer [Access control](#).

- 17. InvocationTargetException

submit command submit task times exception:

```

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 . java : 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

Reason: Check Configuration item workingdir in `conf / sys . properties`

Whether to configure, configure correctly, and verify that the configuration file path is the correct path.

- 18. Do you support setting up agents?

This feature is not supported.

- 19. Why is it expensive for OSS to migrate to OSS?

Refer to [endpoint](#) The domain name in the help, After configuring the internal network domain name, will not charge the traffic fee, but the cost of the number of visits is still charging.

- 20. The synchronization process shows that the source file does not exist

Reason: The Master first lists the list of files, and then moves the data according to the list of files. When list When you finish, certain files on the source end are deleted, you will find that the source file does not exist. This type of file is skipped and output to the error list.

- 21. Turn on incremental mode, will the OSS be deleted after locally deleted?

Turns on incremental mode, if the OSSS is deleted after local deletion, the delete operation is not synchronized.

- 22. Turn on incremental mode, some new documents are not synchronized

The incremental mode uses the last modification of the contrast file to determine whether the file is incremental. Some operations of the file system won't modify the last modified time of objects, such as cp and mv in Windows, and mv and rsync with `- t` or `- a` options in Linux. Data changes from these operations are not detected or synchronized to the OSS.

- 23. The number of tasks shooting the migration has always shown 0

Reasons: again, the more complex, mainly divided into two situations:

- `[2016 - 07 - 21 10 : 21 : 46] [INFO] [name = YoupaiList , totalReque st = 1729925 , avgLatency = 38 ,`

```
recentLate ncy = 300000 ]
```

This log, if the `recentLate ncy = 300000`, is generally normal. List, beat list is slow, usually run up to 30 seconds of timeout, 30 seconds to list out a few files to return a few files, such as the case slowly list tasks It is normal to come out;

- The `recentLate ncy` is very small, and the general case is that the account password is wrong, and so on, because another error in the SDK returns only null), Does not return the error result, so you can only get another error code that is returned by catching the package.
- 24. What do `srcAccessKey`, `srcSecretKey` and `fig` fill in again during the migration

Fill in the operator's account number and password. .

- 25. HTTP is always displayed during another shot migration Error 429

Also shot to limit the SDK access interval, if the access is a little faster, it will limit the speed, please contact us again for Customer Service Release restrictions.

Ossimport itself will try this situation again.

- 26. The execution of Unknown command "Java", Unknown command "nohup" and so on.

Reason: The command used is not installed, please use yum or `apt - get` or

`zypper` Wait for the command to install the corresponding command.

- 27. Task does not match configuration file

The job configuration file appears to be correct, but running looks pretty different from the job profile configuration. Only `sys . properties` properties Changes and then reboots to take effect, and once the job's configuration file is submitted, the modification does not take effect and is required Clean drops the original job, and then resubmits the new configuration file.

- 28. The bucket name "xxx/xx" is invalid

Log reporting exception:

```
java . lang . IllegalArg umentExcep tion : The bucket name
" xxx / xx " is invalid . A bucket name must : 1 ) be
  comprised of lower - case characters , numbers or dash
(-); 2 ) start with lower case or numbers ; 3 ) be
between 3 - 63 characters long .
```

Reason: check if the `destBucket` configuration item (s) are filling correctly, and the bucket is not carrying / and other paths.

- 29. com.aliyun.oss.ClientException: Unknown

Log reporting exception:

```
com . aliyun . oss . ClientExce ption : Unknown
[ ErrorCode ]: NonRepeata bleRequest
[ RequestId ]: Cannot retry request with a non -
repeatable request entity . The cause lists the
reason the original request failed .
```

As well as, usually when the network is full, ossimport will try again, if you still fail after retrying, you can call after the task is complete The retry command retries again.

- 30. Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 timed out

Log reporting exception:

```
Unable to execute HTTP request : Connect to xxx . oss
- cn - beijing - internal . aliyuncs . com : 80 timed out
[ ErrorCode ]: Connection Timeout
[ RequestId ]: Unknown
```

Reason: Non-ECS machines cannot use the internal domain name.

- 31. The specified bucket is not valid

Log reporting exception:

```
com . aliyun . oss . OSSExcepti on : The specified bucket
is not valid .
[ ErrorCode ]: InvalidBuc ketName
[ RequestId ]: 57906B4DD0 EBAB0FF553 D661
[ HostId ]: you - bucket . you - bucketoss - cn - hangzhou -
internal . aliyuncs . com
```

Reason: From the configuration file The destDomian configured domain name cannot have a bucket name.

- 32. Can the srcPrefix in the configuration file specify a file individually?

No, srcPrefix only supports directories or prefix levels, A single file upload can be done with other, simpler tools.

- 33. Unable to execute HTTP request: The Difference between ... is too large.

Log reporting exception:

```
Unable to execute HTTP request : The Difference
between the request time and the current time is
too large .
[ ErrorCode ]: RequestTim eTooSkewed
```

```
[ RequestId ]: xxxxxxxx
```

Reason:

- The Local Machine Time is not good, with a difference of more than 15 minutes from the server time, which is mostly the case.
 - It may be that the concurrency is too high, especially for high CPU usage, leading to slow upload during concurrency.
- 34. No route to host

An error is shown in the logs: `No route to host`. This is probably caused by network interruptions due to a local firewall or iptables.

- 35. Unknown http list file format

The error is displayed using the http mode log because the specified HTTP list file is not in the right format:

- One reason is that the files may be copied from another system. You can use the `mac2unix` or `doc2unix` command to convert the file formats.
 - There are some rows in the file that do not meet the rules, such as a row with fewer than two columns.
- 36. The boject key `"/xxxxx.jpg"` is invalid

Log reporting exception:

```
Exception : java . lang . IllegalArg umentExcep tion : The
boject key "/ xxxxx . jpg " is invalid . An object
name should be between 1 - 1023 bytes long when
encoded as UTF - 8 and cannot contain LF or CR os
unsupporte d chars in XML1 . 0 , and cannot begin
with "/" or "\".
```

Reason:

- Checks whether the `srcPrefix` is as a directory but does not end in;
- Check that the `destPrefix` starts with/or.

5 RAM Policy Editor

Address

[RAM Policy Editor](#)

Usage

RAM authorization policies are composed of several rules. Using the RAM policy editor, you can add or delete rules one by one in the interface, and then a JSON file is automatically generated for the policy. After adding all the policy rules, copy the JSON file and paste it in the created authorization policy content box on the Access Control console.

For detailed operation, see [Create an authorization policy](#).

In the RAM policy editor, you must set these fields for each rule: Effect, Actions, Resources, and Conditions.

- Effect

Specify whether access to this rule is allowed or denied.

- Actions

Specify resource access actions. You can select one or more actions. Generally, it is sufficient to use the wildcard action provided for users:

- `oss :*`: allows all actions
- `oss : Get *` allows all read actions
- `oss : Put *` allows all write actions

For more information, see [RAM Policy Editor README](#).

- Resources

Specify the resources of the OSS authorized to access. You can specify multiple ones, and each would be represented in the following format:

- A bucket: `my - bucket` (with no permission on objects in the bucket)
- All objects in a bucket: `my - bucket /*` (with no permission on the bucket itself, such as ListObjects)
- A directory in a bucket: `my - bucket / dir` (with no permission on objects under dir/)
- All objects under a directory in a bucket: `my - bucket / dir /*` (with no permission on dir, such as ListObjects)
- Complete resource path: `acs : oss :* : 1234 : my - bucket / dir , 1234`
`1234` is the user ID (viewed in the console)

EnablePath

When you want to grant permissions to a directory, you usually need to grant the List permission on its upper level directory. For example, if you want to grant read and write permissions to `my - bucket / users / dir /*`, you also need to grant the following permissions so as to view this directory in the console (or in other tools):

```
ListObject s my - bucket
ListObject s my - bucket / users
ListObject s my - bucket / users / dir
```

When the EnablePath option is selected, the preceding permissions are automatically added.

- Conditions

Specify the conditions that must be met for authorized access. You can specify multiple ones.

For more information, see [RAM Policy Editor README](#).

Example

To grant all permissions for `my - bucket` and its files:

RAM Policy Editor v1.1.0

Chinese Star 2

Add Rules

Effect

Actions

Resources

- After entering a resource name , **press Enter** for confirmation
- Example: my-bucket, my-bucket/dir/
- [More...](#)

EnablePath Automatically add the parent directory permissions ?

Conditions (Optional)

Authorization Policy

```
{
  "Version": "1",
  "Statement": []
}
```

List of Rules

Effect	Actions	Resources	Conditions
--------	---------	-----------	------------

For more examples, see [RAM Policy Editor README](#).

6 ossftp

6.1 Quick installation for OSS FTP

Introduction

The OSS FTP is a special FTP server that maps the operations on files and folders into your OSS instance upon receiving a common FTP request. This utility allows you to use the FTP protocol to manage files stored on your OSS instance.



Note:

OSS SDK is designed for the production environment, and OSS FTP is mainly for individual users.

- Key features
 - **Cross-Platform:** This utility can run on Windows, Linux, and Mac operating systems, either 32 or 64 bit, either on a graphic or command-line interface.
 - **Free of Installation:** You can run this utility directly after extraction.
 - **Free of Configuration:** You can run the utility without any further configurations.
 - **Transparent:** The FTP utility was written in Python, so you can see the complete source code. We will soon make the open source available on GitHub.
- Key functions
 - Supports file/folder upload, download, delete, and other operations
 - Supports multipart upload of large files
 - Supports most FTP commands and can satisfy daily needs



Note:

- Currently, for the ease of installation and deployment, OSS FTP V1.0 does not support TLS encryption. The FTP protocol implements plaintext transmission. To prevent password leaks, we recommend that you run the FTP server and client on the same machine and access using 127.0.0.1:port.
- The utility does not support rename and move operations.

- Do not include any Chinese characters in the extract-to path of the installation package.
- The FTP server's management control page may fail to be opened on early IE browsers.
- Supported Python versions: Python 2.6 and Python 2.7

Downloads

- Windows: [ossftp-1.0.3-win.zip](#)

Now that Python 2.7 is not installed on Windows by default, it is contained in the installation package and is ready for use after extraction, without the hassle of installation and configuration.

- Linux/Mac: [ossftp-1.0.3-linux-mac.zip](#)

Because Python 2.7 or Python 2.6 is installed on Linux and Mac systems by default, the installation packages for Linux and Mac do not contain an executable Python program, but only relevant dependent libraries.

Running

First, extract the downloaded file. Then, select an appropriate running mode based on environmental conditions.

- Windows: Double-click start.vbs to run it.
- Linux: Start the terminal and run it.

```
$ bash start . sh
```

- Mac: Double-click start.command or run it on a terminal.

```
$ bash start . command
```

The preceding process starts an FTP server, which listens to port 2048 at 127.0.0.1 by default. In addition, for ease of control over the status of the FTP server, the program also activates a web server, which listens to port 8192 at 127.0.0.1. If your system has a graphic interface, the control page is automatically opened.



Note:

In most situations, you do not need to configure any settings before running the FTP server. If you make any configuration, remember to restart it to make the changes take effect.

Connecting to the FTP Server

We recommend using the [FileZilla Client](#) to connect to the FTP server. After download and installation, connect to the FTP server as follows:

- Host: 127.0.0.1
- Logon type: normal
- User: access_key_id/bucket_name
- Password: access_key_secret



Note:

- The slash sign (/) means that both, not either items are required. For example, the user could be `tSxyixxxxx xwPMEp / test - hz - jh - 002`.
- For more information about access_key_id and access_key_secret, see [OSS Access Control](#).

Advanced use

- Manage the ftpserver from the console page

- Modify the Listener Address

If you want to access the ftpserver over a network, you must modify the listener address because the default address, 127.0.0.1, only allows local access. You can change it to an intranet IP or Internet IP.

- Modify the Listening Port

Modify the ftpserver' s listening port. We suggest using a port over 1024 because ports below 1024 require administrator permissions.

- Modify the Log Level

Set the ftpserver' s log level. The FTP server' s log is output to the `data / ossftp /` directory. You can view it only by pressing the Log button on the console page. The default log level is INFO and little information is printed in the log. If you need more detailed log information, you can change the level to

DEBUG. If you want to reduce log output, you can set the log level to WARNING or ERROR.

- Set Bucket Endpoints

By default, the ftpserver searches for the bucket's location information, so it can send subsequent requests to the corresponding (such as `oss - cn - hangzhou . aliyuncs . com` or `oss - cn - beijing . aliyuncs . com`).

The ftpserver first tries to access the OSS instance over the intranet. If you set bucket endpoints, for example, `test - bucket - a . oss - cn - hangzhou . aliyuncs . com`, when you access test-bucket-a, you go to the `oss - cn - hangzhou . aliyuncs . com` domain name.

- Set Display Language

By setting cn/en, the display language of the FTP control page can be modified to Chinese/English.



Note:

- The system must be restarted for modifications to take effect.
- All the preceding modifications are actually changes to the ftp directory's config.json file. Thus, you can also modify this file directly.

· Directly start ftpserver (Linux/Mac)

You can only run the ftpserver.py file in the ossftp directory to avoid web_server overhead.

```
$ python ossftp / ftpserver . py &
```

The configuration modification method is the same to the preceding method.

Potential problems

- If you encounter an error when connecting to the FTP server.

The error may be caused by two possible causes:

- There may be an error in the entered `access_key_id` or `access_key_secret`.

Solution: Enter the correct information and try again.

- The used `access_key` information may be a RAM sub-account `access_key` for a sub-account without list buckets permission.

Solution: When using a sub-account, specify bucket endpoints on the console page to tell the ftpserver which endpoint must be used to access a certain bucket. Also, the sub-account must have the required permissions. For information on implementing access control by using RAM to access OSS, see [RAM](#). The details about permissions are as follow:

■ Read-only:

The OSS-FTP must have these permissions: ['ListObjects' , 'GetObject' , 'HeadObject']. For information on creating a RAM sub-account with Read-only permission, see the graphic tutorial [How to Integrate RAM for File Sharing](#).

■ Upload files:

If you want to allow a RAM sub-account to upload files, assign ['PutObject'] permission.

■ Delete files

If you want to allow a RAM sub-account to delete files, assign ['DeleteObject'] permission.

- If you are running the FTP server on Linux, you may encounter the following error when using FileZilla to connect to the server:

```
501 can't decode path (server filesystem encoding
is ANSI_X3.4-1968)
```

This is usually generated when errors occur in local Chinese code. Input the following command in the terminal where you want to run start.sh. Then, restart the program.

```
$ export LC_ALL = en_US . UTF - 8 ; export LANG = " en_US . UTF
- 8 " ; locale
```

6.2 How to store remote attachments to your OSS instance with Discuz

Preface

The website remote attachment function refers to directly storing uploaded attachments to a remote storage server, which is usually a remote FTP server, over the FTP.

Currently, Discuz forums, PHPWind forums, and WordPress websites support the remote attachment function.

This document instructs you on storing remote attachments from a Discuz-based forum.

Preparation

Apply for an OSS account and create a public-read bucket. You must set the permission to public-read because it must allow anonymous access.

Procedures

Here the Discuz version we use is Discuz! X3.1 and the detailed configuration process is shown as follows.

1. Log on to the Discuz website and go to the management interface. Click Global and then Upload Settings.
2. Select Remote Attachments and configure the function.



Note:

- Set “Enable remote attachment” to Yes .

- Set “Enable SSL connection” to No .
- Set the “FTP Server Address” , that is, the address that runs the OSS-FTP. Generally, this is “127.0.0.1 “.
- Set “FTP service port No.” to the default “2048 “.
- Set “FTP Account” in the format of AccessKeyID/BucketName, where “/ “ does not mean “or “.
- Set “FTP Password” to AccessKeySecret.
- Set “Passive Mode Connection” to the default Yes .



Note:

- Set “Remote Attachment Directory” to “. “, that is, to create a directory for upload under the root directory of the bucket.
- Set “Remote URL” to `http :// BucketName . Endpoint` .



Note:

Here, we want to test the bucket test-hz-jh-002 from the Hangzhou region. Therefore, we enter `http :// test - hz - jh - 002 . oss - cn - hangzhou . aliyuncs . com` , where the BucketName must match the endpoint.

- Set the time-out time to 0, that is, to use the default setting of the service.
- After the configuration is complete, click “Test Remote Attachment” . If the test is successful, an information box is displayed.

3. Verification

Ok, now let’ s publish a post on the forum to test the function. On any board, create a post and upload an image as attachment in the post.

Right-click the image and select “Open image in new tab” .

In the browser, you can see the image URL is `http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/forum/201512/18/171012mzvkkku2z3na2w2wa.png`. This indicates that the image has been uploaded to test-hz-jh-002 in the OSS.

6.3 How to store remote attachments to your OSS instance with PHPWind

Preface

The website remote attachment function refers to directly storing uploaded attachments to a remote storage server, which is usually a remote FTP server, over the FTP.

Currently, Discuz forums, PHPWind forums, and WordPress websites support the remote attachment function.

This document instructs you on storing remote attachments from a PHPWind-based forum.

Preparation

Apply for an OSS account and create a public-read bucket. You must set the permission to public-read because it must allow anonymous access.

Procedures

The PHPWind we use is PHPWind 8.7 and the configuration process is as follows.

1. Log on to the website.

Go to the management interface and select Global > Upload Settings > Remote Attachments.

2. Configure the function



Note:

- Set “Enable FTP uploads “ to Yes .
- Set “Website Attachment Address “ to <http://bucket-name.endpoint> Here, we want to test the bucket test-hz-jh-002 from the Hangzhou region. Therefore, we enter <http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com>, where the BucketName must match the endpoint.
- Set the FTP server address, that is, the address that runs the OSS-FTP. Generally , this is 127.0.0.1.
- Set “FTP service port No.” to the default “2048 “.
- Set “Remote attachment directory” to “. “, that is, to create a directory for upload under the root directory of the bucket.

- Set “FTP Account” in the format of AccessKeyID/BucketName, where “/” does not mean “or”.
- Set “FTP Password” to AccessKeySecret. To obtain the AccessKeyID and AccessKeySecret, you can log on to the Alibaba Cloud console and go to Access Key Management.
- Set the FTP time-out time. If you set it to “10”, a time-out response is sent if a request does not receive a response within 10 seconds.

3. Verification

PHPWind does not allow users to directly test the function by clicking a test button. Therefore, we must publish a post with an image to verify the function.

Right-click the image and select “Open image in new tab”. The image is displayed in a new tab.

The image URL indicates that the image has been uploaded to bucket test-hz-jh-002 in the OSS.

6.4 How to store remote attachments to your OSS instance with WordPress

Preface

The website remote attachment function refers to directly storing uploaded attachments to a remote storage server, which is usually a remote FTP server, over the FTP.

Currently, Discuz forums, PHPWind forums, and WordPress websites support the remote attachment function.

This document instructs you on storing remote attachments from a WordPress-based forum.

Preparation

Apply for an OSS account and create a public-read bucket. You must set the permission to public-read because it must allow anonymous access.

Procedures

WordPress does not have inherent support for this function, but implements remote attachment using a third-party plug-in. The WordPress we use is WordPress 4.3.1 and

the plug-in is Hacklog Remote Attachment. The specific configuration process is as follows:

1. Log on to the WordPress website and select “Install Plug-in”. Search for the keyword “FTP” and choose to install Hacklog Remote Attachment.

2. Configuration

- Set the FTP server address, that is, the address that runs the OSS-FTP. Generally, this is 127.0.0.1.
- Set “FTP service port No.” to the default “2048”.
- Set “FTP Account” in the format of AccessKeyID/BucketName, where “/” does not mean “or”.
- Set “FTP Password” to AccessKeySecret.



Note:

To obtain the AccessKeyID and AccessKeySecret, you can log on to the Alibaba Cloud console and go to Access Key Management.

- Set the FTP time-out to the default value, 30 seconds.
- Set “Remote Basic URL” to `http://BucketName.Endpoint/wp`. Here, we want to test the bucket test-hz-jh-002 from the Hangzhou region. Therefore, we enter `http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/wp`.
- Set “FTP Remote Path”. We enter “wp”, that is, to save all attachments to the bucket’s wp directory. Note that this field is related to the “Remote Basic URL” field.
- Set “HTTP Remote Path” to “.”.

For detailed information, see the figure below.

3. Verification

After the configuration is complete, click “Save” and a test starts automatically. The test results are shown at the top of the page.

4. Post a new article and insert an image.

Now you can write a new article and test the remote attachment function. After creating an article, click “Add Media” to upload an attachment.

Upload the attachment as shown in the following figure.

5. When the attachment is uploaded, click “Post” to view your article.

Right-click the image and click “Open image in new tab” to see the image URL.

The image URL indicates that the image has been successfully uploaded to the OSS.

6.5 How to integrate RAM for file sharing

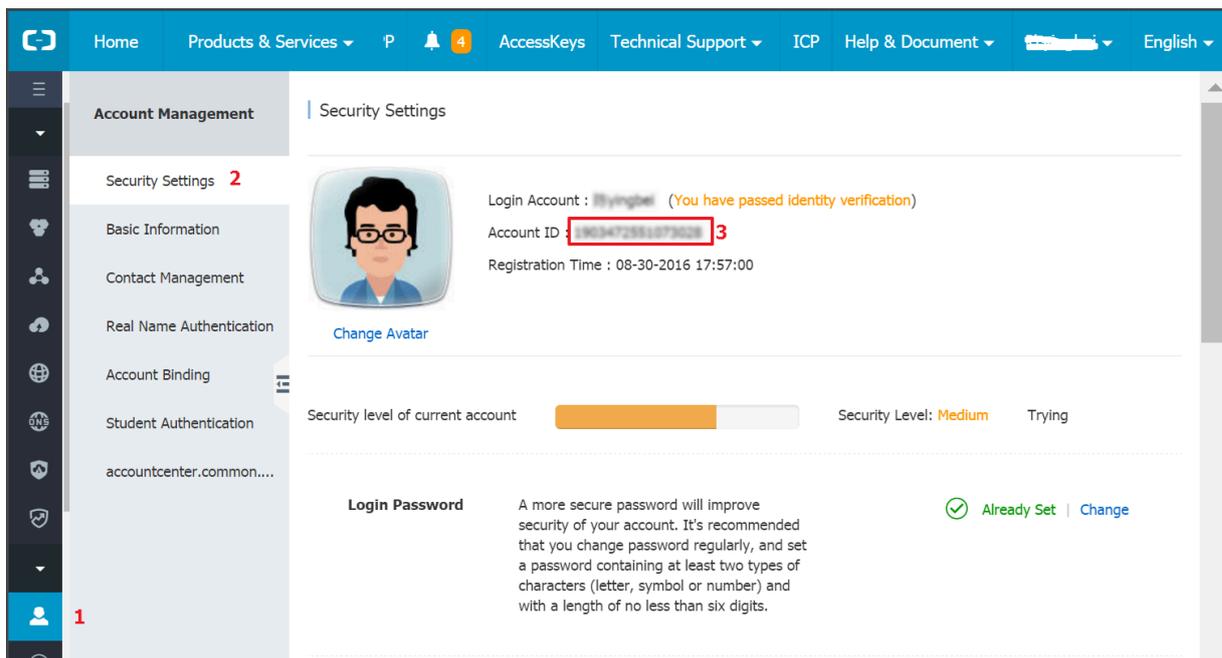
Introduction

This document instructs you on integrating the RAM service to share files and folders in user buckets. Other users have read-only permission, while the bucket owner can edit the objects.

```
Process : Activate RAM -> Create a read - only authorization policy -> Create sub - accounts -> Grant permissions to the sub - accounts -> Verify FTP logon
```

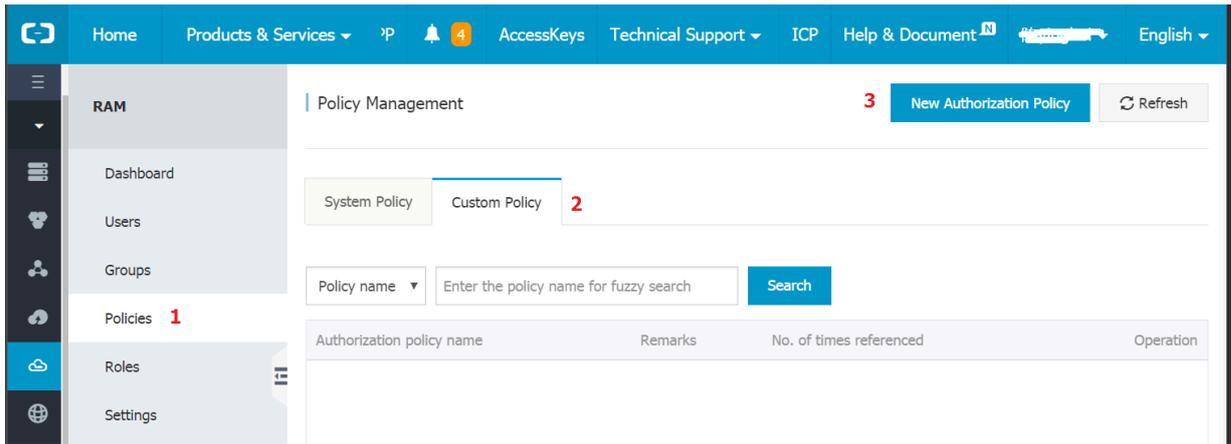
Retrieve account ID

Retrieve your account ID, as shown in the following figure:



Activate RAM

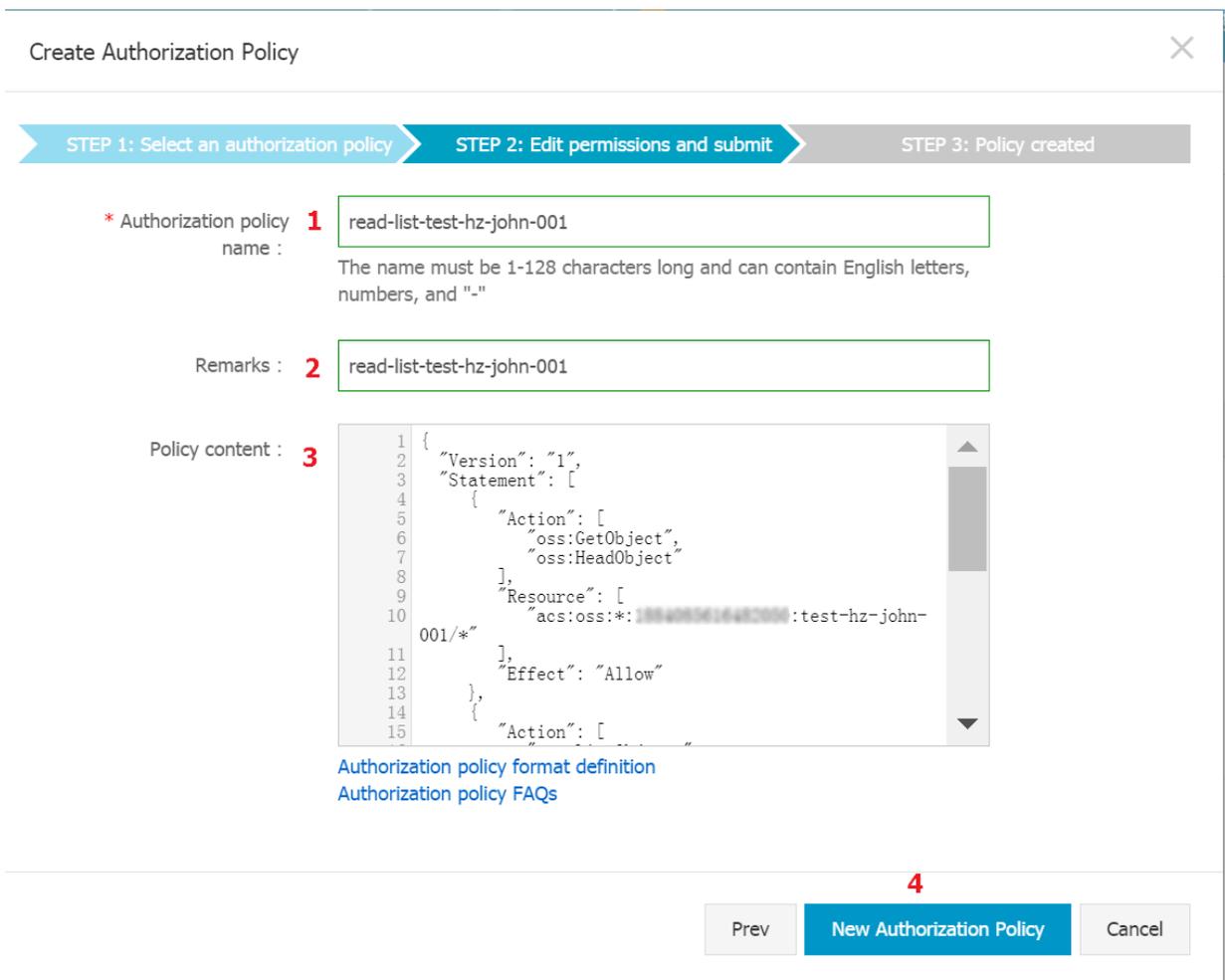
Resource Access Management (RAM) is an Alibaba Cloud service designed for controlling resource access. By creating a policy, you can create a shared read account. Users can use this account to log on to the FTP tool and read your files



Create an authorization policy

After activating RAM, go to the RAM console and click Policies on the left side. Follow the steps shown in the following diagram to create a new authorization policy:

Enter the authorization policy as follows:



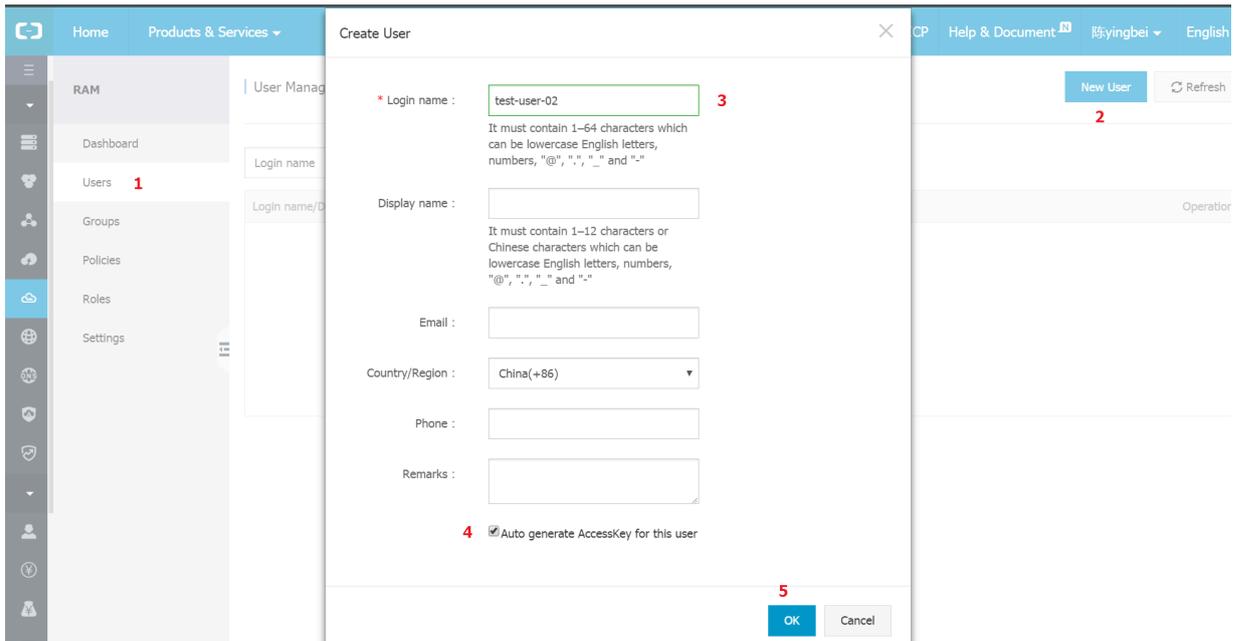
Specify policy name and remarks (fields 1 and 2) as needed. Policy Content in field 3 determines the policy.

```
{
  "Version": " 1 ",
  "Statement": [
    {
      "Action": [
        "oss : GetObject ",
        "oss : HeadObject "
      ],
      "Resource": [
        "acs : oss :*:*****: test - hz - john - 001 /*"
      ],
      "Effect": " Allow "
    },
    {
      "Action": [
        "oss : ListObject s ",
        "oss : GetBucketA cl ",
        "oss : GetBucketL ocation "
      ],
      "Resource": [
        "acs : oss :*:*****: test - hz - john - 001 "
      ],
      "Effect": " Allow "
    },
    {
      "Action": [
        "oss : ListBucket s "
      ],
      "Resource": [
        "acs : oss :*:*****:*"
      ],
      "Effect": " Allow "
    }
  ]
}
```

In the preceding example, replace ***** with your own account ID and replace test - hz - john - 001 with your bucket name. Then, copy all the content and paste it in the policy content. Finally, click New Authorization Policy.

Create an account

The preceding authorization policy produces a read-only policy. Then, we create an account and grant this policy to the account. Follow these steps to create an account:

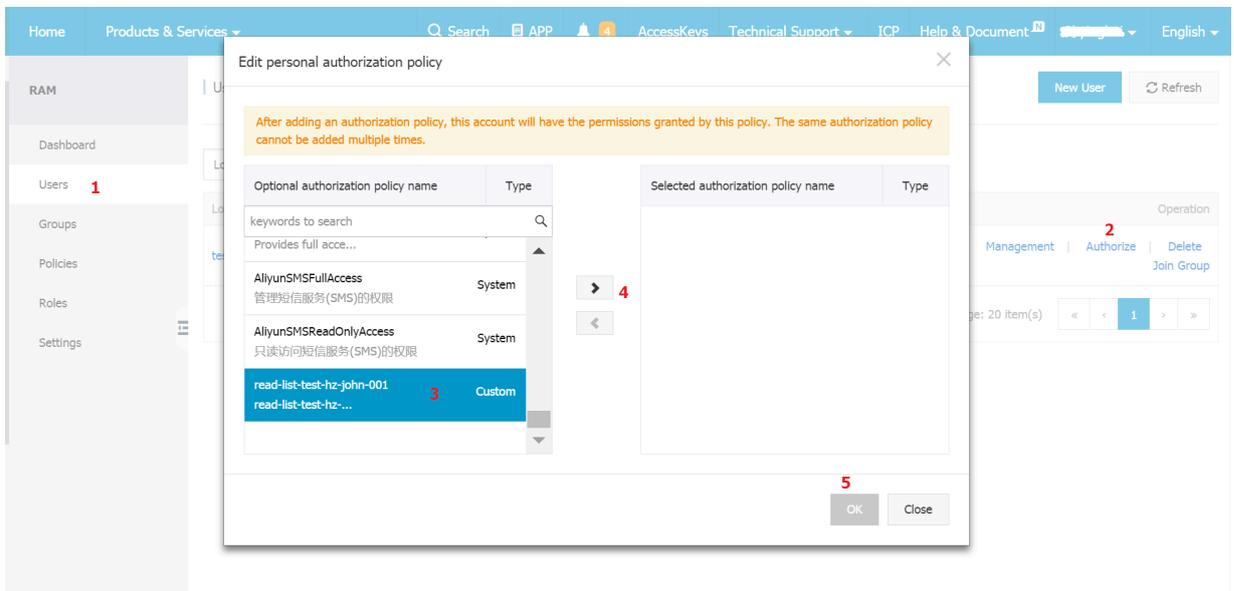


Note:

Remember to record the new account's access_key.

Authorize the account

After that, we grant the new policy to the account.



Log on with the sub-account

Use the sub-account's access_key and the bucket in the authorization policy to log on. Now, you can download files and folders, but upload operations fail.

6.6 FAQ

- Permissions issue:

- Can't list bucket can't log in

The reason is usually the accesskeyid that is used And the accesskeysecret belong to the sub-account, and the sub-account does not have a list Bucket permissions.

If it wasn't in the bucket The Endpoint column configures the bucket's access domain name, And when you access the bucket via ossftp, ftpserver tries to get through Service to get the region of the bucket. At this point if User Account No list The bucket permission causes the login to fail.

The solution is in the bucket. The three-level access domain name that is configured in the endpoints, such.

- List file is reported wrong after login is successful

This is typically the accesskeyid used. And accesskeysecret belong to the sub-account, and the sub-account does not have List objects (equivalent to get Bucket) permissions.

- Other questions:

- List file timeout causes the connection to be disconnected after the login is successful

The reason is generally that there are too many files or folders in the bucket root directory. After logging in to FTP, ftpserver tries to list all the files/folders in the bucket root directory, you can list 1000 files/folders at a time. If there are more than 1 million files/folders in the root directory, this will result in more than 1000 HTTP requests, which can easily lead to a timeout.

- A machine running ftpserver failed data transfer due to a port Restriction

Because the control port of the FTP Protocol differs from the data port, when the ftpserver is working in passive mode, whenever you need to transfer data , ftpserver opens 1 random port, waiting for the client to connect. So when

the ftpserver machine has a port limit, it may cause the data to fail to transfer properly.

The workaround is when running ftpserver. py, by specifying -- Passive_ports_start and -- Passive_ports_end parameter to set the start and end ranges of the local port, and then open the ports for that range.

- The connection between the client and the ftpserver is often disconnected

Each FTP client typically has a timeout setting, which can be set to not time out . Take the filezilla tool for example, in the settings-> connection, you can set the timeout to 0.

7 ossfs

7.1 Quick installation

Ossfs allows you to mount Alibaba Cloud OSS buckets to local files in Linux systems. In the system, you can quickly use the local file system to perform operations on OSS objects, achieving data sharing.



Notice:

Note the following limits when using ossfs:

- If you edit a uploaded file, the file is uploaded again.
- The performance of metadata-related operations, such as `ls` `directory`, is poor because these operations need to access the OSS server remotely.
- An error may occur if you rename an object or a folder. Operation failures may cause inconsistent data.
- ossfs does not apply to scenarios where read and write operations are highly concurrent.
- You must maintain data consistency when a OSS bucket is mounted to multiple clients. For example, you must schedule the usage of an object to prevent it from being written by multiple clients at the same time.
- Hard links are not supported.



Note:

You can use Cloud Storage Gateway (CSG) to access OSS. In this way, OSS buckets are mapped to local directories or disks.

- CSG supports the NFS and SMB (CIFS) protocols so that it can allow you to access shared directories based on OSS.
- CSG also supports the iSCSI protocol. Therefore, it can map massive OSS buckets to local disks and provides efficient elastic storage solution.

Features

Ossfs is constructed based on S3FS and incorporates all S3FS functions, including:

- Supports most functions of the POSIX file system, including file reading/writing, directories, link operations, permissions, UID/GID, and extended attributes.
- Uploads large files using the OSS multipart function.
- Supports MD5 verification which ensures data integrity.

Installation and use

- Installation package download

Released Linux	Download
Ubuntu 16.04 (x64)	ossfs_1.80.5_ubuntu16.04_amd64.deb
Ubuntu 14.04 (x64)	ossfs_1.80.5_ubuntu14.04_amd64.deb
CentOS 7.0 (x64)	ossfs_1.80.5_centos7.0_x86_64.rpm
CentOS 6.5 (x64)	ossfs_1.80.5_centos6.5_x86_64.rpm

Due to the lower version of the Linux distribution, the kernel version is relatively lower. The ossfs is prone to disconnection or other problems during the running process. Therefore, users are advised to upgrade the operating system to CentOS 7.0 or Ubuntu 14.04 or later.

- Installation method

- Run the following commands to install ossfs for Ubuntu:

```
sudo apt - get update
sudo apt - get install gdebi - core
sudo gdebi your_ossfs _package
```

- Run the following command to install ossfs for CentOS 6.5 or later:

```
sudo yum localinstall ll your_ossfs _package
```

- Run the following command to install ossfs for CentOS 5:

```
sudo yum localinstall ll your_ossfs _package --nogpgcheck
```

- Usage

Set bucket name and AccessKeyId/Secret and save it to the /etc/passwd-ossfs file.

Note that the permissions for this file must be set correctly. We suggest setting it to 640.

```
echo my - bucket : my - access - key - id : my - access - key -
secret > / etc / passwd - ossfs
```

```
chmod 640 / etc / passwd - ossfs
```

Mount the OSS bucket to the specified directory.

```
ossfs my - bucket my - mount - point - ourl = my - oss - endpoint
```

Example:

Mount the bucket `my - bucket` to the `/ tmp / ossfs` directory. The AccessKeyId is `faint`, the AccessKeySecret is `123`, and the OSS endpoint is `http :// oss - cn - hangzhou . aliyuncs . com`.

```
echo my - bucket : faint : 123 > / etc / passwd - ossfs
chmod 640 / etc / passwd - ossfs
mkdir / tmp / ossfs
ossfs my - bucket / tmp / ossfs - ourl = http :// oss - cn - hangzhou . aliyuncs . com
```



Notice:

If you use an Alibaba Cloud ECS instance to provide ossfs services, you can use the intranet endpoints. In this example, you can replace the OSS endpoint with `oss - cn - hangzhou - internal . aliyuncs . com` to save bandwidth costs. For more information about intranet endpoints, see [Regions and endpoints](#).

Unmount the bucket:

```
fusermount - u / tmp / ossfs
```

For more information, see [GitHub ossfs](#).

Release log

For more information, see [GitHub ChangeLog](#).

7.2 FAQ

- Q: For what programs is ossfs suitable?
 - ossfs mounts OSS buckets locally. If you want a program that does not support OSS to automatically sync the data to the OSS, ossfs is a great option.
- Q: What are the limitations of ossfs?
 - Because data must be synced to the cloud over the network, the performance and functions of ossfs may differ from those of local file systems. If you want to run a database or other applications with frequent I/O operations on a mounted

ossfs disk, you must consider this carefully. ossfs differs from local file systems in the following ways:

- Random write and append operations overwrite the entire file.
- The performance of metadata operations, such as list directory, is poor because the system has to remotely access the OSS server.
- The file/folder rename operation is not atomic.
- When multiple clients are attached to a single OSS bucket, you must coordinate the actions of each client manually. For example, you must avoid multiple clients writing the same file.
- Hard link is not supported.

- Q: Do I need to use Alibaba Cloud hosts for ossfs?
 - ossfs does not need to be used with Alibaba Cloud intranet. It can be used on external Internet hosts.
- Q: Can ossfs simultaneously mount multiple OSS buckets?
 - Yes, write multiple OSS configuration information entries in the passwd-ossfs file. Buckets from different OSS accounts are supported.
- Q: I installed ossfs at yum/apt-get and has an error: conflicts with file from package fuse-devel.
 - There is an earlier version of fuse on your system. Please use the relevant package manager to uninstall and then reinstall ossfs.
- Q: ossfs is not working properly, how do I debug?
 - You can use the `-d -o f2` parameter when mounting. ossfs will write log content into the system logs. On the centos system, in/var/log/messages.
 - You can also use the `-f -d -o f2` parameter when mounting, and ossfs prints the logs to the screen.
- Q: When trying to mount a bucket, why do I receive the error “ossfs: unable to access MOUNTPOINT /tmp/ossfs: Transport endpoint is not connected” ?
 - First, run the `umount` command for the corresponding directory.
 - When mounting with ossfs, check that the entered URL parameter is correct and the bucket, AccessKey ID, and AccessKey secret match.
 - DO NOT include the bucket name in the URL. For example, if the bucket domain name is `ossfs-test-1.oss-cn-hangzhou.aliyuncs.com` on

the OSS console, set the URL to `http://oss-cn-hangzhou.aliyuncs.com`.

- Q: Why does ossfs display “ossfs: unable to access MOUNTPOINT /tmp/odat: No such file or directory” ?
 - This error occurs if the directory is not yet created. You must create the directory before mounting.
- Q: Why does the “operation not permitted” error occur after I mount the bucket locally and run the ls command for the directory?
 - In your bucket, check if the directory name contains any OSS objects with invisible characters. The file system has strict restrictions for file/directory names. If the directory name fails to meet the restrictions, this error occurs. Use another tool to rename these objects and run the ls command, the directory content can be correctly displayed.
- Q: There are a lot of files in one of my directories. Why is ls so slow?
 - Assuming that there are n files in a directory, then the ls of this directory requires at least a minimum of n oss http requests. When there are many files, this can cause serious performance problems.
 - You can optimize in two ways:
 - Increase stat cache size with the `-omax_stat_cache_size=xxx` parameter, so that the first time ls will be slow, but the subsequent ls will be fast, because the metadata of the file is in the local cache. The default is 1000, which costs about 4 MB of memory, please adjust to the appropriate value according to the size of your machine's memory.
 - Use the `ls -f` command, which eliminates n HTTP requests with OSS.
 - For more information.
- Q: How do I set permissions during ossfs mounting?
 - If you want to allow other users to access mounted folders, specify the `allow_others` parameter as follows when running ossfs:
 - ```
ossfs your_bucket your_mount_point - ourl =
your_endpoint - o
```

```
allow_othe r
```

- Why does the allow\_other parameter still have no access to the file?
    - Note: allow\_other is the permission granted to other users in the Mount directory, not the file inside! If you want to change the files in the folder, use the chmod command.
  - allow\_other gives the Mount directory 777 permission by default, and I want to have the Mount directory permission 770, what should I do?
    - You can set by umask.
  - Q: If you want to allow the mounting of folders (/tmp/ossfs) that belong to another user,
    - Method 1: If you want to allow the mounting of folders (/tmp/ossfs) that belong to another user, you need to create the mount folder as user and use ossfs:
      - `sudo -u user mkdir /tmp/ossfs`
      - `sudo -u user ossfs bucket - name /tmp/ossfs`
    - Method 2: first get the uid/gid information for the specified user by the id command. For example, to get uid/gid information for a www user: `id www`; then specify the uid/gid parameter when you mount:
      - ```
ossfs your_bucket your_mount_point - ourl = your_url
- ouid = your_uid
- ogid = your_gid
```
- Note: uid/gid are numbers.
- Q: I am not the root user, how does umount ossfs mount the directory?
 - `fusermount -u your_mountpoint`
 - Q: How can I mount ossfs automatically when the device starts up?
 - Step 1: Write the bucket name, AccessKeyId/Secret, and other information into /etc/passwd-ossfs, and change the permissions for this file to 640.
 - ```
echo your_bucket_name : your_access_key_id : your_access_key_secret > /etc/passwd-ossfs
```
      - `chmod 640 /etc/passwd-ossfs`

- **Step 2: Make the appropriate settings (the setting methods differ for different system versions).**
  - **Step 2A: Use the fstab method to automatically mount the ossfs (applies to Ubuntu 14.04 and CentOS 6.5).**
    - **Add the following command in /etc/fstab:**
      - ```
ossfs # your_bucket_name your_mount_point fuse
_netdev , url = your_url , allow_other 0
```
 - **In the preceding command, replace ‘your_xxx’ with your actual bucket name and other information.**
 - **Save the /etc/fstab file. Run the `mount -a` command. If no error is reported, the settings are correct.**
 - **Now, Ubuntu 14.04 can automatically mount the ossfs. For CentOS 6.5, also run the following command:**
 - ```
chkconfig netfs on
```
  - **Step 2B: Mount ossfs using a boot script (applies to CentOS 7.0 and later).**
    - **Create the file ossfs in the /etc/init.d/ directory. Copy the content in the [Template File](#) to the new file. Here, replace ‘your\_xxx’ with your own information.**
    - **Run the command: `chmod a+x /etc/init.d/ossfs`.**
    - **The preceding command grants execution permission to the new ossfs script. You can now run this script. If no errors occur in the script content, the OSS bucket has been mounted to the specified directory.**
    - **Run the command: `chkconfig ossfs on`.**
    - **The preceding command sets the ossfs boot script as another service, so it is automatically started when the device starts up.**
    - **ossfs can now automatically mount upon startup. To sum up, if you use Ubuntu 14.04 or CentOS 6.5, perform Steps 1 and 2A; if you use CentOS 7.0, perform Steps 1 and 2B.**
- **Q: How do I solve the fusermount: failed to open current directory: Permission denied error?**
  - **This is a fuse bug. It requires the current user to have read permission for the current directory (unmounted directory). To solve this problem, run the `cd`**

command to change to a directory with read permission and then run the ossfs command again.

- Q: I need to use a www user to mount ossfs. In this case, how do I set up automatic mounting?
  - See the answer to the preceding question. Perform Step 1 as stated. Perform Step 2B with the command in the /etc/init.d/ossfs file changed to:

```
sudo -u www ossfs your_bucket your_mount_point - ourl
= your_url
```

- Set the boot script to allow the use of sudo to edit /etc/sudoers. Change the Defaults requiretty line to # Defaults requiretty (comment out this line).
- Q: How do I solve the fusemount : failed to open current directory : Permission denied error?
  - This is a [fuse bug](#). It requires the current user to have read permission for the current directory (unmounted directory). To solve this problem, run the cd command to change to a directory with read permission and then run the ossfs command again.
- Q: How do I avoid the cost of scanning files by using ECS to mount ossfs?
  - The program scans a directory mounted by ossfs to convert to a request to OSS, if the number of requests is high, costs will be incurred (1 cent/10 thousand times ). If it is [updatedb](#), you can skip it by modifying /etc/updatedb.conf. The specific practice is:
    1. Add fuse . ossfs to PRUNEFS =.
    2. Add the mounted directory to the PRUNEPATHS =.
  - How do I determine which process swept my catalog?
    1. First install auditd: sudo apt-get install auditd.
    2. Start auditd: sudo service auditd start.
    3. Set the monitor mount directory : auditctl -w /mnt/ossfs
    4. In the auditorium log, you can see which processes have accessed this directory: ausearch -i | grep /mnt/ossfs

- **Q: what is the content-type file that uses ossfs to upload to OSS all "application/ocdet-stream? what happened?**
  - ossfs queries /etc/mime.types content to determine the Content-Type of the file, please check that the file exists, if it does not exist, you need to add:
    1. For Ubuntu, you can add it with `udo apt-get install mime-support`.
    2. For Centos, you can add it with `sudo Yum install mailcap`
    3. You can also manually add one row per format, each in the form of:  
 Application/JavaScript JS
- **Q: How do I start ossfs using the supervisor?**
  1. To install the supervisor, run the `sudo apt-Get install supervisor` in Ubuntu
  2. Create a directory and edit the ossfs STARTUP script:

```
mkdir / root / ossfs_scri pts
vi / root / ossfs_scri pts / start_ossf s . sh
```

Write the following data:

```
Unload
fusermount -u /mnt/ossfs
Re-mounted, you must add -F parameter to run
ossfs, let ossfs run at the front desk
exec ossfs my - bucket my - mount - point - ourl = my - oss
- endpoint - f
```

3. Edit/etc/Supervisor/supervisord.conf to add the following paragraph at the end:

```
[program : ossfs]
command = bash / root / ossfs_scri pts / start_ossf s . sh
logfile = / var / log / ossfs . log
log_stdout = true
log_stderr = true
logfile_maxbytes = 1MB
logfile_backups = 10
```

4. Run Supervisor:

```
supervisor d
```

**supervisord**

5. Confirm that everything is fine:

```
ps aux | grep supervisor # should be able to see
the supervisor Process
ps aux | grep ossfs # should be able to see
ossfs Process
kill -9 ossfs # Kill ossfs process, the supervisor
must restart it, do not use killall, because
killall sends sigterm, the process Exits normally,
and the Supervisor no longer reruns ossfs.
```

```
ps aux | grep ossfs # should be able to see ossfs
Process
```

If an error occurs, check `/var/log/supervisor/supervisord.log` and `/var/log/ossfs.log`.

- Q: encounter "fuse: Warning: Library too old, some operations may not work?"

This occurs because of the libfuse version that ossfs uses at compile time Higher than the libfuse version linked to at run time. This is often due to the user's own installation of libfuse. Install ossfs with the RPM package we provide, without having to install libfuse again.

The RPM bag that we provide on the box and the box contains the box, if there is a chain in the running environment and ossfs is linked to an earlier version of fuse, the preceding warning will appear.

### 1. How do I confirm the fuse version of The ossfs runtime link?

- Run `LDD $(which ossfs) | grep Fuse`
- For example, the result is `"/lib64/libfuse. So. 2 "`, then you can see the version of fuse through `LS-L/lib64/libfuse`.

### 2. How do I link ossfs to the correct version?

- First find the directory of libfuse with `rpm-QL ossfs | grep fuse`.
- For example, the result is `"/usr/lib/libfuse. So. 2 "`, use `fig =/usr/lib ossfs... Run ossfs`

### 3. Can I ignore this warning?

- You better not see this bug.
- Q: Why do I see file information with ossfs (for example, size) not consistent with what other tools see?

Because ossfs, by default, caches the file's meta-information (including size/permissions, etc ), this does not require every time ls requests are sent to OSS to speed up. If the user passes other programs (such as SDK/website console/osscommand, etc) the file has been modified so that it is possible to see the file information in ossfs, not updated in a timely manner.

If you want to disable ossfs caching, you can add the following paramete -

```
omax_stat_ cache_size = 0
```

# 8 osscmd (unavailable)

---

## 8.1 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.

## 8.2 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 AccessKey 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_endpo int
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_endpo int
```

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 getallbuck et
```

No buckets are displayed if the user has no buckets in OSS.

- Create a bucket

Create a bucket named mybucketname.

```
$ python osscmd createbuck et mybucketna me
```

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 getallbuck et
```

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 :// mybucketna me /
```

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_exis ted_file 7625e1adc3 a4b129763d
580ca0a78e 44 local_exis ted_file
$ python osscmd put local_exis ted_file oss ://
mybucketna me / test_objec t
```



Note:

`md5sum` runs in Linux only.

- View objects again

You can view objects that have been uploaded to a bucket.

```
$ python osscmd list oss :// mybucketna me /
```

- 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 :// mybucketna me / test_objec t
download_f ile
$ md5sum download_f ile
7625e1adc3 a4b129763d 580ca0a78e 44 download_f ile
```



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 deletelife cycle oss :// mybucket
0 . 139 (s) elapsed
```

- **Read a lifecycle rule.**

```
python osscmd getlifecyc le oss :// mybucket
Error Headers :
[(' content - length ', ' 288 '), (' server ', ' AliyunOSS '),
 (' connection ', ' close '), (' x - oss - request - id ', '
54C74FEE5D 7F6B24E504 2630 '), (' date ', ' Tue , 27 Jan
2015 08 : 44 : 30 GMT '), (' content - type ', ' applicatio n
/ xml ')]
Error Body :
<? xml version =" 1 . 0 " encoding =" UTF - 8 "? >
< Error >
 < BucketName > mybucket </ BucketName >
 < Code > NoSuchLife cycle </ Code >
 < Message > No Row found in Lifecycle Table . </ Message
 >
 < RequestId > 54C74FEE5D 7F6B24E504 2630 </ RequestId >
 < HostId > mybucket . oss - maque - hz - a . alibaba . net </
HostId >
</ Error >
Error Status :
404
getlifecyc le Failed !
```

## Configure hotlinking protection

- **Allow access from a request that has an empty Referer field.**

```
$ osscmd putreferer oss :// test -- allow_empt y_referer =
true
0 . 004 (s) elapsed
```

- **Obtain the Referer whitelist.**

```
$ osscmd getreferer oss :// test
<? xml version =" 1 . 0 " encoding =" UTF - 8 "? >
< RefererCon figuration >
 < AllowEmpty Referer > true </ AllowEmpty Referer >
 < RefererLis t />
</ RefererCon figuration >
```

- **The Referer field is required. Only requests that have the Referer field value of test are allowed.**

```
$ osscmd putreferer oss :// test -- allow_empt y_referer =
false -- referer =' www . test . com '
0 . 092 (s) elapsed
```

- **Obtain the Referer whitelist.**

```
$ osscmd getreferer oss :// test
<? xml version =" 1 . 0 " encoding =" UTF - 8 "? >
```

```
< RefererCon figuration >
 < AllowEmpty Referer > false </ AllowEmpty Referer >
 < RefererLis t >
 < Referer > www . test . com </ Referer >
 </ RefererLis t >
</ RefererCon figuration >
```

- **The Referer field is required. Only requests that have the Referer field value of test or test1 are allowed.**

```
$ osscmd putreferer oss :// test -- allow_empt y_referer =
false -- referer =' www . test . com , www . test1 . com '
```

- **Obtain the Referer whitelist.**

```
$ osscmd getreferer oss :// test
<? xml version =" 1 . 0 " encoding =" UTF - 8 "? >
< RefererCon figuration >
 < AllowEmpty Referer > false </ AllowEmpty Referer >
 < RefererLis t >
 < Referer > www . test . com </ Referer >
 < Referer > www . test1 . com </ Referer >
 </ RefererLis t >
</ RefererCon figuration >
```

## Use logging

- **Configure a logging rule**

```
$ osscmd putlogging oss :// mybucket oss :// myloggingb ucket
/ mb
```

- **Obtain logging rules that are configured for a bucket**

```
$ osscmd getlogging oss :// mybucket
```

## 8.3 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:

```
getallbuck et (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 getallbuck et`
- `python osscmd gs`

### createbucket(cb,mb,pb)

#### Command:

```
createbuck et (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 createbuck et oss :// mybucket`
- `python osscmd cb oss :// myfirstbuc ket -- acl = public - read`
- `python osscmd mb oss :// mysecondbu cket -- acl = private`
- `python osscmd pb oss :// mythirdbuc ket`

### deletebucket(db)

#### Command:

```
deletebuck et (db) oss :// bucket
```

Delete a bucket. `db` is short for delete bucket.

**Example:**

- `python osscmd deletebucket et 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:**

```
putlifecyc le 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 putlifecyc le oss :// mybucket lifecycle . xml
```

**Example:**

```
< LifecycleC onfigurati on >
 < Rule >
 < ID > 1125 </ ID >
 < Prefix > log_backup </ Prefix >
 < Status > Enabled </ Status >
 < Expiration >
 < Days > 2 </ Days >
 </ Expiration >
 </ Rule >
</ LifecycleC onfigurati on >
```

## getlifecycle

**Command:**

```
osscmd getlifecyc le oss :// bucket
```

Obtain lifecycle rules of a bucket.

**Example:**

```
python osscmd getlifecyc le oss :// mybucket
```

## deletelifecycle

**Command:**

```
osscmd deletelife cycle oss :// bucket
```

Delete all lifecycle rules of a bucket.

**Example:**

```
python osscmd deletelife cycle 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.test1.com, www.test2.com` 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 . test1 . com , www . test2 . com "
```

## 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
```

## 8.4 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:**

```
listallobj ect 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 listallobj ect oss :// mybucket`
- `python osscmd listallobj ect oss :// mybucket / testfolder /`

**deleteallobject****Command:**

```
deleteallo bject 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 /`

### downloadalobject

**Command:**

```
downloadalobject 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 downloadalobject oss :// mybucket / tmp / folder`
- `python osscmd downloadalobject oss :// mybucket / tmp / folder -- replace = false`
- `python osscmd downloadalobject 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 downloadto dir oss :// mybucket / tmp / folder`
- `python osscmd downloadto dir oss :// mybucket / tmp / folder`  
`-- replace = false`
- `python osscmd downloadto dir oss :// mybucket / tmp / folder`  
`-- replace = true`

#### uploadfromdir

##### Command:

```
uploadfrom dir localdir oss :// bucket / [prefix] --
check_poin t = check_poin t_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_poin t = check_poin t_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 uploadfrom dir / mytemp / folder oss :// mybucket`
- `python osscmd uploadfrom dir / mytemp / folder oss :// mybucket -- check_poin t_file = / tmp / mytemp_rec ord . txt`
- `python osscmd uploadfrom dir C :\ Documents and Settings \ User \ My Documents \ Downloads oss :// mybucket -- check_poin t_file = C :\ cp . txt`

### put

#### Command:

```
put localfile oss :// bucket / object -- content - type = [
content_ty pe]
-- 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_ty pe]
-- 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_t ype = 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_t ype = url`

### signurl(sign)

#### Command:

```
signurl (sign) oss :// bucket / object -- timeout =[timeout_se
conds]
```

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`

## 8.5 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 =
75835E389E A648C0B935 71B6A46023 F3
```

## 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:**

```
getallpart size oss :// bucket
```

List the total size of parts that are uploaded by using the existing upload IDs.

**Example:**

```
python osscmd getallpart size 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
=
```

```
D9D278DB6F 8845E9AFE7 97DD235DC5 76
```

## 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 multiupload / 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 = D9D278DB6F 8845E9AFE7 97DD235DC5 76`
- `python osscmd multiupload / tmp / localfile . txt oss :// mybucket / object`

```

-- thread_num = 5
python osscmd mp / tmp / localfile . txt oss :// mybucket /
object -- max_part_n um = 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:

```

uploadpart fromfile (upff) localfile oss :// bucket / object
-- upload_id = xxx
-- part_number = xxx

```

This command is used for tests only.

## uploadpartfromstring(upfs)

### Command:

```

uploadpart fromstring (upfs) oss :// bucket / object --
upload_id = xxx -- part_number = xxx
-- data = xxx

```

This command is used for tests only.