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

6. Please contact Alibaba Cloud directly if you discover any errors in this document.
## Document conventions

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal disclaimer</td>
<td>I</td>
</tr>
<tr>
<td>Document conventions</td>
<td>I</td>
</tr>
<tr>
<td>1 OSS tools</td>
<td>1</td>
</tr>
<tr>
<td>2 ossbrowser</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Quick start</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Permission management</td>
<td>10</td>
</tr>
<tr>
<td>3 ossutil</td>
<td>14</td>
</tr>
<tr>
<td>3.1 Overview</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Download and installation</td>
<td>16</td>
</tr>
<tr>
<td>3.3 Common commands</td>
<td>18</td>
</tr>
<tr>
<td>3.3.1 appendfromfile</td>
<td>18</td>
</tr>
<tr>
<td>3.3.2 bucket-encryption</td>
<td>20</td>
</tr>
<tr>
<td>3.3.3 bucket-policy</td>
<td>22</td>
</tr>
<tr>
<td>3.3.4 bucket-tagging</td>
<td>25</td>
</tr>
<tr>
<td>3.3.5 bucket-versioning</td>
<td>27</td>
</tr>
<tr>
<td>3.3.6 cat</td>
<td>29</td>
</tr>
<tr>
<td>3.3.7 config</td>
<td>30</td>
</tr>
<tr>
<td>3.3.8 cors</td>
<td>34</td>
</tr>
<tr>
<td>3.3.9 cors-options</td>
<td>36</td>
</tr>
<tr>
<td>3.3.10 cp</td>
<td>38</td>
</tr>
<tr>
<td>3.3.11 create-symlink</td>
<td>55</td>
</tr>
<tr>
<td>3.3.12 du</td>
<td>57</td>
</tr>
<tr>
<td>3.3.13 getallpartsize</td>
<td>59</td>
</tr>
<tr>
<td>3.3.14 hash</td>
<td>60</td>
</tr>
<tr>
<td>3.3.15 help</td>
<td>62</td>
</tr>
<tr>
<td>3.3.16 lifecycle</td>
<td>62</td>
</tr>
<tr>
<td>3.3.17 listpart</td>
<td>66</td>
</tr>
<tr>
<td>3.3.18 logging</td>
<td>67</td>
</tr>
<tr>
<td>3.3.19 ls</td>
<td>69</td>
</tr>
<tr>
<td>3.3.20 mb</td>
<td>75</td>
</tr>
<tr>
<td>3.3.21 mkdir</td>
<td>78</td>
</tr>
<tr>
<td>3.3.22 object-tagging</td>
<td>79</td>
</tr>
<tr>
<td>3.3.23 probe</td>
<td>83</td>
</tr>
<tr>
<td>3.3.24 read-symlink</td>
<td>88</td>
</tr>
<tr>
<td>3.3.25 referer</td>
<td>90</td>
</tr>
<tr>
<td>3.3.26 restore</td>
<td>92</td>
</tr>
<tr>
<td>3.3.27 request-payment</td>
<td>94</td>
</tr>
<tr>
<td>3.3.28 rm</td>
<td>96</td>
</tr>
<tr>
<td>3.3.29 set-acl</td>
<td>100</td>
</tr>
<tr>
<td>3.3.30 set-meta</td>
<td>103</td>
</tr>
</tbody>
</table>
# 1 OSS tools

In addition to the OSS console, you can use the following tools to streamline OSS operations.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ossbrowser</strong></td>
<td>A graphical object management tool.</td>
</tr>
<tr>
<td></td>
<td>• Provides an easy-to-use graphical interface.</td>
</tr>
<tr>
<td></td>
<td>• Provides features similar to those of Windows Explorer.</td>
</tr>
<tr>
<td></td>
<td>• Allows you to browse objects.</td>
</tr>
<tr>
<td></td>
<td>• Allows you to upload and download directories (folders).</td>
</tr>
<tr>
<td></td>
<td>• Allows you to use concurrent upload and resumable upload to upload objects.</td>
</tr>
<tr>
<td></td>
<td>• Allows you to configure authorization policies and grant permissions to RAM users.</td>
</tr>
<tr>
<td></td>
<td>• Supports Windows, Linux, and macOS.</td>
</tr>
<tr>
<td><strong>Limits:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The transmission speed and performance of ossbrowser are not as good as those of ossutil because ossbrowser is a graphical tool.</td>
</tr>
<tr>
<td></td>
<td>• Only objects smaller than 5 GB can be uploaded or replicated.</td>
</tr>
<tr>
<td></td>
<td>• Objects larger than 48.8 TB cannot be uploaded.</td>
</tr>
<tr>
<td><strong>ossutil</strong></td>
<td>A command line tool used to manage objects and buckets.</td>
</tr>
<tr>
<td></td>
<td>• Provides a wide range of simple and convenient commands to manage objects and buckets while ensuring high operation performance.</td>
</tr>
<tr>
<td></td>
<td>• Allows you to use concurrent upload and resumable upload to upload objects.</td>
</tr>
<tr>
<td></td>
<td>• Allows you to upload and download directories (folders).</td>
</tr>
<tr>
<td>Tool</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>osscmd (unavailable)</strong></td>
<td>A command line tool used to manage objects and buckets.</td>
</tr>
<tr>
<td></td>
<td>• Provides a wide range of commands to manage objects and buckets.</td>
</tr>
<tr>
<td></td>
<td>• Supports Windows and Linux.</td>
</tr>
<tr>
<td></td>
<td><strong>Limits:</strong></td>
</tr>
<tr>
<td></td>
<td>• osscmd is compatible with Python 2.5, 2.6, and 2.7 only. It is not compatible with Python 3.x.</td>
</tr>
<tr>
<td></td>
<td>• osscmd does not allow you to configure new features such as the storage class of Infrequent Access (IA) or Archive, cross-region replication (CRR), and back-to-origin.</td>
</tr>
</tbody>
</table>

⚠️ **Notice:**
Commands supported by osscmd have been integrated into **ossutil**. osscmd is no longer available for downloads as of July 31, 2019.
<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ossfs</strong></td>
<td>A tool used to attach a bucket to the local file system.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>• Supports most functions provided by POSIX-based file systems, including file read/write, directories, symbolic links, permissions, UIDs or GIDs, and extended attributes.</td>
</tr>
<tr>
<td></td>
<td>• Allows you to use multipart upload to upload large objects.</td>
</tr>
<tr>
<td></td>
<td>• Supports MD5 verification to ensure data integrity.</td>
</tr>
<tr>
<td></td>
<td><strong>Limits:</strong></td>
</tr>
<tr>
<td></td>
<td>• You cannot attach a bucket of the Archive storage class.</td>
</tr>
<tr>
<td></td>
<td>• If you use ossfs to edit an uploaded object, the object is uploaded again.</td>
</tr>
<tr>
<td></td>
<td>• The performance of metadata-related operations, such as list directory, is compromised because you must remotely access the OSS server.</td>
</tr>
<tr>
<td></td>
<td>• Errors may occur when you rename an object or a folder. Operation failures may cause data inconsistency.</td>
</tr>
<tr>
<td></td>
<td>• ossfs is not suitable for scenarios that require highly concurrent read and write operations.</td>
</tr>
<tr>
<td></td>
<td>• If an OSS bucket is attached to multiple clients, you are responsible for maintaining data consistency. For example, you must schedule when users can use objects to prevent multiple clients from writing to the same object at the same time.</td>
</tr>
<tr>
<td></td>
<td>• Hard links are not supported.</td>
</tr>
<tr>
<td><strong>ossftp</strong></td>
<td>An FTP-based tool used to manage objects in OSS.</td>
</tr>
<tr>
<td></td>
<td>• You can use FTP clients such as FileZilla, WinSCP, and FlashFXP to manage objects in OSS.</td>
</tr>
<tr>
<td></td>
<td>• ossftp is an FTP server that receives FTP requests and performs operations on objects and folders in OSS.</td>
</tr>
<tr>
<td></td>
<td>• ossftp is based on Python 2.7 and later.</td>
</tr>
<tr>
<td></td>
<td>• Supports Windows, Linux, and macOS.</td>
</tr>
<tr>
<td>Tool</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ossimport</strong></td>
<td>A tool used to synchronize data to OSS.</td>
</tr>
<tr>
<td></td>
<td>• Allows you to synchronize data from a third-party data source to OSS.</td>
</tr>
<tr>
<td></td>
<td>• Supports distributed deployment. You can use multiple servers to migrate data simultaneously.</td>
</tr>
<tr>
<td></td>
<td>• Supports migrating TB-grade data volumes and above.</td>
</tr>
<tr>
<td></td>
<td>• Supports Windows and Linux.</td>
</tr>
<tr>
<td></td>
<td>• Applicable to Java 1.7 and 1.8.</td>
</tr>
<tr>
<td><strong>RAM Policy Editor</strong></td>
<td>A tool used to achieve automatic generation of OSS-related permission policies.</td>
</tr>
<tr>
<td></td>
<td>• Generates permission policies based on specified information. You can add the generated policies to the custom policy in the RAM console.</td>
</tr>
<tr>
<td></td>
<td>• Supports the following browsers: Google Chrome, Firefox, and Safari.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>We recommend that you use this tool to generate custom authorization policies.</td>
</tr>
<tr>
<td><strong>oss-emulator</strong></td>
<td>A lightweight simulator of OSS.</td>
</tr>
<tr>
<td></td>
<td>• Provides the same APIs as those of OSS. oss-emulator can be used to debug and test functions.</td>
</tr>
<tr>
<td></td>
<td>• oss-emulator is based on Ruby 2.2.8 and later.</td>
</tr>
<tr>
<td></td>
<td>• Supports Windows and Linux.</td>
</tr>
</tbody>
</table>
2 ossbrowser

2.1 Quick start

ossbrowser is a graphical management tool developed by Alibaba Cloud with similarity to Windows Explorer. You can use ossbrowser to browse, upload, download, and manage objects.

Precautions

- You can only upload objects smaller than or equal to 48.8 TB in size by using ossbrowser.
- You can only move or copy objects smaller than or equal to 5 GB in size by using ossbrowser. For objects larger than 5 GB, we recommend that you use ossutil.
- ossbrowser is supported on Linux, macOS, and Windows 7 or later versions. We recommend that you do not use ossbrowser on Windows XP or Windows Server.

Quick installation

1. Download and install ossbrowser.

<table>
<thead>
<tr>
<th>Supported operating system</th>
<th>Download URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows x32</td>
<td>Windows x32</td>
</tr>
<tr>
<td>Windows x64</td>
<td>Windows x64</td>
</tr>
<tr>
<td>macOS</td>
<td>macOS</td>
</tr>
<tr>
<td>Linux x64</td>
<td>Linux x64</td>
</tr>
</tbody>
</table>

Note:
For more download URLs, visit GitHub.
2. Start and log on to ossbrowser.

- **Endpoint**: Select the endpoint that you want to access.
  - Default: Log on to ossbrowser with the default endpoint.
  - Customize: Enter the endpoint to be used to log on to ossbrowser. You can use a URL starting with "http" or "https" to log on to ossbrowser over HTTP or HTTPS. Example: https://oss-cn-beijing.aliyuncs.com. For more information about regions and endpoints, see #unique_11.
  - **cname**: You can log on to ossbrowser with a custom domain name (CNAME) bound to your OSS resources. For more information about how to bind custom domain names, see #unique_12.

- **AccessKeyId** and **AccessKeySecret**: Enter the AccessKey pair of your account. To secure data, we recommend that you log on to ossbrowser by using the
AccessKey pair of a RAM user. For more information about how to obtain AccessKey pairs, see #unique_13.

- **Preset OSS Path:**
  - If you log on to ossbrowser as an administrator RAM user that has administrative permissions on all buckets, you do not need to configure this parameter.
  - If you log on to ossbrowser as an operator RAM user, you must configure this parameter. Enter the path of the OSS bucket or subdirectory that you want to access (the RAM user must have the permissions to access the OSS bucket or subdirectory). The path is in the following format: `oss://bucketname/path`.
  - request payer: Select this check box if pay-by-requester is enabled for the corresponding path. For more information about pay-by-requester, see #unique_14.
  - Region: Select the region to which the OSS resources belong.

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

Manage buckets

- **Create a bucket**
  1. On the homepage of ossbrowser, click Create a bucket.
  2. Configure bucket information.
    - Name: The name of a bucket can be up to 63 characters in length. It must be unique.
    - Region: Set the region for the bucket.
    - ACL: Set ACL for the bucket. For more information about ACL, see Access control based ACLs.
    - Type: Set the default storage class for the bucket. For more information about storage classes, see #unique_16.
  3. After the configuration is complete, click OK.
Object Storage Service

Tools / 2 ossbrowser

- Delete a bucket

Select the bucket that you want to delete, and choose More > Remove. A bucket cannot be deleted while objects or parts are stored within it.

Manage objects

- Create a directory

1. On the homepage of ossbrowser, click the bucket in which you want to create a directory.
2. Click Directory.
3. Set the directory name and click OK.

Note:
- The directory name must be UTF-8 characters and cannot contain emoticons.
- 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 subdirectory cannot contain two consecutive periods .. within its name.
- The directory name must be 1 to 254 characters in length.

- Upload objects or folders

In the specified bucket or directory, click Files or Folder. Select the objects or folders that you want to upload. You can upload multiple objects or folders at a time.

- Download objects or folders

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

- Copy objects or folders

1. In the specified bucket or directory, select the object or folder that you want to copy, and click Copy.
2. Go to the bucket or directory to which you want to copy the data, and click Paste. If the source and destination addresses 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 object has been stored for less
than the specified number of days, early deletion fees are incurred. For more information, see *Billing items and methods*.

- **Move objects or folders**

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

  2. Go to the bucket or directory to which you want to copy the data, and click Paste.

  ![Notice]

  When you move an object or a folder, the object or folder is copied from the source address to the destination address, and the object or folder in the source address is deleted. If the storage class of the moved object is IA or Archive and the object has been stored for less than the specified number of days, early deletion fees are incurred.

- **Rename objects or folders**

  In the specified bucket or directory, select the object or folder than you want to rename, and choose More > Rename. Enter the new name.

  ![Notice]

  - You can only rename objects smaller than 1 GB.
  - When you rename an object or a folder, the object or folder is copied, renamed, and then saved. The original object or folder is deleted. If the storage class of the renamed object is IA or Archive and the object has been stored for less than the specified number of days, early deletion fees are incurred.

- **Delete objects or folders**

  Select the object or folder that you want to delete, and choose More > Remove. If the storage class of the deleted object is IA or Archive and the object has been stored for less than the specified number of days, early deletion fees are incurred.
· Generate a URL for an object

1. Select the specified object, and choose More > Address.
2. Set the validity period of the URL, and click Generate.
3. Click Copy or Mail it to send the URL to users who want to access the object.
   The generated QR code can also be used to access the object.

· Preview objects

You can double-click an object to preview it. Ossbrowser allows you to preview objects in TXT format and image objects smaller than 5 MB.

· Manage parts

Select the specified bucket, and click Multipart. You can delete parts that you no longer use.

2.2 Permission management

This topic describes how to perform simple permission management through Ossbrowser.

Log on to Ossbrowser as a RAM user

For data security, we recommend that you use the AccessKey pair of a RAM user to log on to Ossbrowser.

Note:

Create a RAM user and an AccessKey pair. For more information, see Create a RAM user.

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

· Administrator RAM user: a RAM user with administrative 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 user, as shown in the following figure.

- Operator RAM user: a RAM user with the read-only permission on a bucket or directory. Administrator RAM users can use the simple policy function to authorize RAM users. For more information, see the Grant permissions with a simple policy section.

Note:
You can grant fine-grained permissions to RAM users. For more information, see Overview.

Log on to ossbrowser with STS tokens

You can use an STS token to log on to ossbrowser. STS tokens can be given to other authorized users for temporary access to your bucket directory. The STS token automatically becomes invalid after it expires.

1. Log on to ossbrowser as an administrator RAM user.

Notice:
When you log on to ossbrowser with your Alibaba Cloud account or as an administrator RAM user, some features are inaccessible to ensure data security. Use the AccessKey pair of an administrator RAM user to log on to ossbrowser and generate a token. The administrator RAM user must have the permissions to manage a bucket or directory, manage RAM (AliyunRAMFullAccess), and call the STS AssumeRole operation (AliyunSTSAssumeRoleAccess).

2. Select the objects or directories to be accessed temporarily by the authorized users, and choose More > Authorization Token, as shown in the following figure.

3. Save the obtained token.

4. Log off from ossbrowser and use the STS token to log on, as shown in the following figure.
Grant permissions with a simple policy

After logging on to ossbrowser as an administrator RAM user, you can use the Simplify Policy function to create an operator RAM user, or grant an operator RAM user the read-only or read/write permissions on a bucket or directory.

Note:
The simple policy function of ossbrowser is designed based on Alibaba Cloud RAM to control access. You can log on to the RAM console through the Alibaba Cloud website to manage your RAM users more precisely.

1. Log on to ossbrowser as an administrator RAM user.
2. Select one or more objects or directories to be accessed temporarily by the authorized users, and choose More > Simple Policy.
3. In the Simplify policy authorization dialog box that appears, set Privileges.
4. Grant permissions to an existing operator RAM user or create a new operator RAM user in this dialog box.

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

Notice:
To use the simple policy function, you must log on to ossbrowser by using the AccessKey pair of a RAM user that has the RAM configuration permissions. For example, use the AccessKey pair of an administrator RAM user that has the RAM configuration permissions.
3 ossutil

3.1 Overview

ossutil allows you to use command lines to manage OSS data, and also provides a variety of simple commands to manage buckets and objects. The following operating systems support ossutil: Windows, Linux, and macOS.

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 Download and installation.

Common commands

The following table describes common ossutil commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>appendfromfile</td>
<td>Uploads a local file to an appendable object in OSS.</td>
</tr>
<tr>
<td>bucket-encryption</td>
<td>Adds, modifies, queries, or deletes encryption configurations for a bucket.</td>
</tr>
<tr>
<td>bucket-policy</td>
<td>Adds, modifies, queries, or deletes the bucket policy for a bucket.</td>
</tr>
<tr>
<td>bucket-tagging</td>
<td>Adds, modifies, queries, or deletes tagging configurations for a bucket.</td>
</tr>
<tr>
<td>bucket-versioning</td>
<td>Adds or queries versioning configurations for a bucket.</td>
</tr>
<tr>
<td>cat</td>
<td>Exports object content to ossutil.</td>
</tr>
<tr>
<td>config</td>
<td>Generates a configuration file to store OSS access information.</td>
</tr>
<tr>
<td>cors</td>
<td>Adds, modifies, queries, or deletes cross-origin resource sharing (CORS) configurations for a bucket.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cors-options</td>
<td>Tests whether a bucket allows a specified CORS request.</td>
</tr>
<tr>
<td>cp</td>
<td>Uploads, downloads, or copies objects.</td>
</tr>
<tr>
<td>create-symlink</td>
<td>Creates a symbolic link.</td>
</tr>
<tr>
<td>du</td>
<td>Obtains the storage size of a specified bucket, object, or object directory.</td>
</tr>
<tr>
<td>getallpartsize</td>
<td>Obtains the size of each part in a bucket that has not been uploaded and the total size of all of those parts.</td>
</tr>
<tr>
<td>hash</td>
<td>Calculates the CRC-64 or MD5 value of a local file.</td>
</tr>
<tr>
<td>help</td>
<td>Obtains help information about a command. We recommend that you use the <code>help</code> command to obtain information about a specified command.</td>
</tr>
<tr>
<td>lifecycle</td>
<td>Adds, modifies, queries, or deletes lifecycle configurations for a bucket.</td>
</tr>
<tr>
<td>listpart</td>
<td>Lists the parts that have not been uploaded for an object.</td>
</tr>
<tr>
<td>logging</td>
<td>Adds, modifies, queries, or deletes logging configurations for a bucket.</td>
</tr>
<tr>
<td>ls</td>
<td>Lists buckets, objects, or parts.</td>
</tr>
<tr>
<td>mb</td>
<td>Creates a bucket.</td>
</tr>
<tr>
<td>mkdir</td>
<td>Creates a directory in a bucket.</td>
</tr>
<tr>
<td>object-tagging</td>
<td>Adds, modifies, queries, or deletes tagging configurations for an object.</td>
</tr>
<tr>
<td>probe</td>
<td>Monitors access to OSS, and troubleshoots problems caused by network faults or incorrect parameter settings during the upload and download process.</td>
</tr>
<tr>
<td>read-symlink</td>
<td>Reads the description of a symbolic link object.</td>
</tr>
<tr>
<td>referer</td>
<td>Adds, modifies, queries, or deletes hotlink protection configurations for a bucket.</td>
</tr>
<tr>
<td>restore</td>
<td>Restores an object from the frozen state to the readable state.</td>
</tr>
<tr>
<td>request-payment</td>
<td>Configures or queries pay-by-requester configurations for a bucket.</td>
</tr>
<tr>
<td>rm</td>
<td>Deletes buckets, objects, or parts.</td>
</tr>
<tr>
<td>set-acl</td>
<td>Configures the ACL for a bucket or an object.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>set-meta</td>
<td>Configures the metadata for an uploaded object.</td>
</tr>
<tr>
<td>sign</td>
<td>Generates signed URLs for third-party users to access objects in a bucket.</td>
</tr>
<tr>
<td>stat</td>
<td>Obtains the description of a specified bucket or object.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the ossutil version.</td>
</tr>
<tr>
<td>website</td>
<td>Adds, modifies, queries, or deletes static website hosting and back-to-origin configurations for a bucket.</td>
</tr>
</tbody>
</table>

3.2 Download and installation

This topic describes how to download and install ossutil.

Version and runtime environment

- Current version: 1.6.10
- Source code: ossutil
- Runtime environment
  - Windows, Linux, or macOS
  - Support architectures: x86 (32-bit and 64-bit) and ARM (32-bit and 64-bit)

Download URLs

- Linux x86 32-bit
- Linux x86 64-bit

Notice:
When you copy the URLs into the wget command to download ossutil, delete the ? spm=xxxx section from the URLs.

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

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

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

  1. Download the ossutil installation package:

     ```
     wget http://gosspublic.alicdn.com/ossutil/1.6.10/ossutil64
     ```

  2. Modify the file execution permissions:

     ```
     chmod 755 ossutil64
     ```

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

     ```
     ./ossutil64 config
     ```

     Enter the name of the configuration file. The file name can contain a path. The default path is /home/user/.ossutilconfig. If you press Enter without specifying a different destination, the file will be generated in the default path. If you want to generate the file in another path, set the --config-file option to the path.

     If the path of the configuration file is not entered, the default configuration file /home/user/.ossutilconfig 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.

     - **endpoint**: specifies the domain name of the region to which the bucket belongs. For more information, see [Regions and endpoints](#unique_57).

     - **accessKeyID**: For more information about how to view the AccessKey ID, see [unique_57](#unique_57).

     - **accessKeySecret**: For more information about how to view the AccessKey secret, see [unique_57](#unique_57).

     - **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](#unique_57).

     Note:

     For more information about the configuration file, see `config`. 
• Install ossutil in Windows (the 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 (the 64-bit macOS system is used as an example)
  1. Download the ossutil installation package.

    curl -o ossutilmac64 http://gosspublic.alicdn.com/ossutil/1.6.6/ossutilmac64

  2. Modify the file execution permissions:

    chmod 755 ossutilmac64

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

./ossutilmac64 config

3.3 Common commands

3.3.1 appendfromfile

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

Note:
For more information about append upload, see #unique_60.

Command syntax

    ./ossutil appendfromfile 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 `set-meta` command to modify the metadata of an object.

Examples

Upload an object by using append upload.

```
./ossutil appendfromfile /file/test.txt oss://bucket1/test.txt
```

```text
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 `appendfromfile` command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--meta</code></td>
<td>Configures the metadata of a specified object. The <code>--meta</code> option cannot be used to append content to an object that already exists.</td>
</tr>
<tr>
<td><code>--maxupspeed</code></td>
<td>Specifies the maximum upload speed. Unit: KByte/s. Default value: 0 (unlimited).</td>
</tr>
<tr>
<td><code>--encoding-type</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: info: generates prompt logs, debug: generates detailed logs that contain HTTP requests and responses.</td>
</tr>
<tr>
<td><code>--payer</code></td>
<td>Specifies the payer of the request. You can set this option to requester to enable the pay-by-requester mode.</td>
</tr>
<tr>
<td><code>--proxy-host</code></td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 proxies are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.<strong>:</strong>:1080.</td>
</tr>
<tr>
<td><code>--proxy-user</code></td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

Note:
For more information about common options, see View all supported options.

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_62.

Command syntax

- Add or modify bucket encryption configurations

  ```bash
  ./ossutil bucket-encryption --method put oss://bucket --sse-algorithm algorithmName [--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-6eab1ea****
  ```

• 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--sse-algorithm</code></td>
<td>Specifies the server-side encryption algorithm. Valid values: KMS and AES256.</td>
</tr>
<tr>
<td><code>--kms-masterkey-id</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>--method</code></td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• put: adds or modifies bucket encryption configurations.</td>
</tr>
<tr>
<td></td>
<td>• get: obtains bucket encryption configurations.</td>
</tr>
<tr>
<td></td>
<td>• delete: deletes bucket encryption configurations.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.0.0:3128">http://120.79.0.0:3128</a> or socks5://120.79.0.0:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#).

### 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_63](#).

**Command syntax**

- Add or modify bucket policy configurations

  ```bash
  ./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:

  ```json
  {
    "Version": "1",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": ["ram:ListObjects"
      ],
      "Principal": []
  }
  ```
• 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:

```json
{
    "Version": "1",
    "Statement": [
        {
            "Action": ["oss:*"],
            "Effect": "Allow",
            "Principal": ["*"],
            "Resource": ["acs:oss::*:174649585760****:bucket1",
                         "acs:oss::*:174649585760****:bucket1/*"],
            "Condition": {"IpAddress": {"acs:SourceIp": ["10.10.10.10"]}}
        }
    ]
}
```
• Obtain bucket policy configurations

```bash
./ossutil.exe bucket-policy --method get oss://bucket1
{
    "Version": "1",
    "Statement": [
        {
            "Action": ["oss:*"],
            "Effect": "Allow",
            "Principal": ["*"]
        },
        "Resource": [
            "acs:oss::*:174649585760****:bucket1",
            "acs:oss::*:174649585760****:bucket1/*"
        ],
        "Condition": {
            "IpAddress": {
                "acs:SourceIp": ["10.10.10.10"]
            }
        }
    ]
}
```

• Delete bucket policy configurations

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

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>--method</strong></td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>· put: adds or modifies bucket policy configurations.</td>
</tr>
<tr>
<td></td>
<td>· get: obtains bucket policy configurations.</td>
</tr>
<tr>
<td></td>
<td>· delete: deletes bucket policy configurations.</td>
</tr>
<tr>
<td><strong>--loglevel</strong></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>· info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>· debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="https://120.79.***:3128">https://120.79.***:3128</a> or socks5://120.79.***:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#).

### 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 [#unique_64](#).

**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**

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

**Delete bucket tagging configurations**

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

**Examples**

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

- **Query bucket tagging configurations**
  ```bash
  ./ossutil bucket-tagging --method get oss://bucket1
  ```

- **Delete bucket tagging configurations**
  ```bash
  ./ossutil bucket-tagging --method delete oss://bucket1
  ```

**Common options**

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--method</td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• put: adds or modifies bucket tagging configurations.</td>
</tr>
<tr>
<td></td>
<td>• get: obtains bucket tagging configurations.</td>
</tr>
<tr>
<td></td>
<td>• delete: deletes bucket tagging configurations.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.***:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**

For more information about common options, see [View all supported options](#).

### 3.3.5 bucket-versioning

The **bucket-versioning** command is used to configure versioning for a bucket or query versioning configurations.

**Note:**

For more information about the command used to enable versioning on a bucket, see #unique_65.

**Command syntax**

- Enable or suspend versioning for a bucket

  ```
  ./ossutil bucket-versioning --method put oss://bucket versioning _parameter
  ```

  When the `--method` option is set to `put`, the `versioning_parameter` parameter can only be set to **enabled** or **suspended**.

  - Versioning is enabled when the `versioning_parameter` parameter is set to enabled,
  - Versioning is suspended when the `versioning_parameter` parameter is set to suspended.
• Query versioning configurations for a bucket

./ossutil bucket-versioning --method get oss://bucket

Examples

• Enable versioning for a bucket

./ossutil bucket-versioning --method put oss://bucket1 enabled

• Suspend versioning for a bucket

./ossutil bucket-versioning --method put oss://bucket1 suspended

• Query versioning configurations for a bucket

./ossutil bucket-versioning --method get oss://bucket1
bucket versioning status:Suspended

Note:
A bucket can be in any of the following versioning states:

- Enabled: indicates that versioning is enabled for the bucket. When an object in this state is deleted or overwritten, an object version is generated.
- Suspended: indicates that versioning is suspended for the bucket. In this state, previously generated versions of an object are preserved as historical versions. When the object is overwritten or deleted again, a version that has a version ID of null is generated, and the newly generated null version overwrites the existing one.
- Null: indicates that versioning has never been enabled for the bucket. If versioning has been enabled for the bucket, the bucket versioning state can only be Enabled or Suspended, and cannot be changed to Null.

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--method</td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• put: enables or suspends versioning.</td>
</tr>
<tr>
<td></td>
<td>• get: obtains versioning configurations.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***.3128">http://120.79.***.3128</a> or socks5://120.79.***:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#).

### 3.3.6 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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>* info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>* debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--payer</td>
<td>Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.**.*:3128">http://120.79.**.*:3128</a> or socks5://120.79.**.*:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#).

### 3.3.7 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
  Enter the name of the configuration file. The file name can contain a path, and the default path is /home/user/.ossutilconfig. If you press Enter without specifying a different destination, the file will be generated in the default path. If you want to generate the file in another path, set the --config-file option to the path. If the path of the configuration file is not entered, the default configuration file /home/user/.ossutilconfig is used. The following parameters are ignored if you press Enter without configuring them. To obtain 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](#unique_57).
  - **accessKeyID**: For more information about how to view the AccessKey ID, see [unique_57].
  - **accessKeySecret**: For more information about how to view the AccessKey secret, see [unique_57].
  - **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](#unique_57).

- Generate a configuration file in non-interactive mode

  ```
  ./ossutil config -e oss-cn-beijing.aliyuncs.com -i LTAIbZcdVCmQ****
  -k D26oqKBudxDRBg8Wuh2EWDBrM0**** -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:

```plaintext
[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 accelerated domain name or canonical domain name for each specified bucket. If you configure the Bucket-Cname option, ossutil searches for the accelerated domain name specified for a bucket when you perform operations on the bucket. If the specified accelerated domain name exists, ossutil replaces the endpoints specified in the Bucket-Endpoint and Credentials options with the domain name. For more information about canonical domain names, see the "Configure a canonical name (CNAME) record" section in Overview.

- **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 only need 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_67.

**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

---

Notice:
take effect in the configuration file. You can specify an endpoint or accelerated domain name for each bucket in the configuration file.

- ossutil can specify endpoints for multiple regions. The priorities of all of the 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-e, --endpoint</td>
<td>Specifies the endpoint in the [Credentials] section of the configuration file.</td>
</tr>
<tr>
<td>-i, --access-key-id</td>
<td>Specifies the AccessKey ID in the [Credentials] section of the configuration file.</td>
</tr>
<tr>
<td>-k, --access-key-secret</td>
<td>Specifies the AccessKey secret in the [Credentials] section of the configuration file.</td>
</tr>
<tr>
<td>-t, --sts-token</td>
<td>Optional. This option specifies the STS token in the [Credentials] section of the configuration file.</td>
</tr>
<tr>
<td>--output-dir</td>
<td>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. The default value is the ossutil_output directory in the current directory.</td>
</tr>
<tr>
<td>-L, --language</td>
<td>Specifies the language ossutil uses. Valid values: CH and EN. Default value: CH. To set this option to CH, make sure that your system supports UTF-8 encoding.</td>
</tr>
</tbody>
</table>
### 3.3.8 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_68](#unique_68).

#### Command syntax

- **Add or modify CORS configurations for a specified bucket**

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

Tools / ossutil

- Obtain CORS configurations for a bucket

```bash
./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

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

Examples

- Add CORS configurations for a bucket

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

- Obtain CORS configurations for a bucket

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

- Delete CORS configurations for a bucket

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

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--method</code></td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>- put: adds or modifies CORS configurations.</td>
</tr>
<tr>
<td></td>
<td>- get: obtains CORS configurations.</td>
</tr>
<tr>
<td></td>
<td>- delete: deletes CORS configurations.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>- info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>- debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td><code>--proxy-host</code></td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.<strong>:</strong>:1080.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#unique_68).

### 3.3.9 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_68.
- For more information about commands used to configure CORS, see `cors`.

**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

```bash
./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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--origin</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>--acr-method</code></td>
<td>Specifies the value of the Access-Control-Request-Method field in the HTTP header. Valid values: GET, PUT, POST, DELETE, and HEAD.</td>
</tr>
<tr>
<td><code>--acr-headers</code></td>
<td>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 (&quot;). Example: <code>--acr-headers &quot;header1,header2,header3.&quot;</code></td>
</tr>
<tr>
<td><code>--encoding-type</code></td>
<td>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.</td>
</tr>
</tbody>
</table>
| `--loglevel`      | Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:  
  - `info`: generates prompt logs.  
  - `debug`: generates detailed logs that contain corresponding HTTP request and response information. |
<p>| <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. |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**

For more information about common options, see [View all supported options](#).

### 3.3.10 cp

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

**Command syntax**

- **Upload objects**

  ```bash
  ```

- **Download objects**

  ```bash
  ```

- **Copy objects**

  ```bash
  ```

**Examples for object uploads**

- **Upload a single object**

  ```bash
  ./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 `header:value#header:value...` format. Run the following command to upload the `a.txt` object and configure its metadata:

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

**Note:**
For more information about how to configure metadata for uploaded objects, see `set-meta`.

- **Upload a folder**

  You can add the `-r` option to the `cp` command to upload a folder to OSS.

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

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

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

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

  When uploading an object, you can use the `--maxupspeed` option to configure the maximum upload speed measured in KB/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 MB/s
    ```bash
don./ossutil cp a.jpg oss://bucket/path --maxupspeed 1024
  ```
  
  - Upload a folder and set the maximum upload speed to 1 MB/s
    ```bash
don./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` and `--exclude` option to the `cp` command to specify the conditions for selecting objects.

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

  - `*`: matches all characters. For example, `*.txt` specifies any TXT objects.
  - `?`: matches any single character. For example, `abc?.jpg` specifies 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]. The objects include `abc1.jpg`, `abc2.jpg`, `abc3.jpg`, `abc4.jpg`, and `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 `abc8.jpg`.

Note:
- Names that include directories, such as `--include "/usr/test/".jpg"`, are not supported.
- To specify the `--include` and `--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 excluded because it is in TXT format. The final matching results exclude the `test.txt` object.

- **Rule 2:** `--exclude "*.txt" --include "*test*"`. When ossutil reads the `--exclude "*.txt"` condition, the `test.txt` object is excluded. When ossutil reads the `--include "*test*"` condition, the `test.txt` object matches the condition because its name contains test. The final matching results include the `test.txt` object.

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

Examples:

- Upload all objects that are in 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 JPG or TXT format.
  ```
  ./ossutil cp dir/ oss://my-bucket/path --include "*abc*" --exclude "*.jpg" --exclude "*.txt" -r
  ```

• Upload an object and configure its storage class

When uploading an object, you can use the `--meta` option to configure 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 and store the encrypted object in a bucket. For more information about server-side encryption, see #unique_62.

- 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 amount of fees 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 in the destination bucket or those 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 the 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 pay-by-requester enabled

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

- Upload only objects in the current directory and ignore subdirectories

  ./ossutil cp dir/ oss://bucket1/path --only-current-dir -r

- Do not generate an object for an uploaded directory

  ./ossutil cp dir/ oss://bucket1/path --disable-dir-object -r

The OSS directory is simulated with an object that is 0 KB in size and whose name ends with a forward slash (/). If you specify the --disable-dir-object option when uploading objects, ossutil does not generate an object for the directory to which the objects belong. However, you can see the corresponding directory structure in the OSS console. If you delete all objects from the directory, the directory is also deleted.
- **upload objects in the subdirectory to which the symbolic link points**
  
  
  ```bash
  ./ossutil cp dir/ oss://bucket1/path --enable-symlink-dir -r
  ```

  - **All objects in the subdirectory to which the symbolic link points and the subdirectory to which the symbolic link points are ignored during object uploads.**

  ```bash
  ./ossutil cp dir/ oss://bucket1/path -r --disable-all-symlink
  ```

Examples for object downloads

- **Download a single object**

  ```bash
  ./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 as a file to your local device.

  ```bash
  ./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 pay-by-requester enabled**

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

- **Download a folder**

  ```bash
  ./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 in the destination bucket or those have a later last modification time than that of the destination objects. The command is as follows:

  ```bash
  ./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`.

```bash
./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` and `--exclude` to specify the conditions for selecting objects. For more information, see the `--include` and `--exclude` section in this topic.

- Download all objects that are not in JPG format

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

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

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

• Download a specified version of an object from a versioning-enabled bucket

```bash
./ossutil cp oss://bucket1/test.jpg dir/ --version-id CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjlhYzQzZjNlYTAYZDE3MDRk
```

To use the `--version-id` option, you must run the `ls --all-versions` command to obtain version IDs of the object.

Note:
The `--version-id` option can only be used for objects in versioning-enabled buckets. For the command used to enable versioning on a bucket, see `bucket-versioning`. 
• Download only objects in the current directory and ignore subdirectories
  
  ./.ossutil cp oss://bucket1/path dir/ --only-current-dir -r

Examples for copying 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 --include and --exclude to specify the conditions for selecting objects. For more information, see the --include and --exclude section in this topic.

  - Copy all objects that are not in 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 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
    
  
  - Set the storage class of all objects 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 `restore` command to restore the object to the readable state, and then use the `cp` command to modify its storage class.

When you use the `cp` command to overwrite an object, it is considered overwritten and 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 and methods*.

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

You can specify the server-side encryption method when you upload an object and store the encrypted object in a bucket. For more information about server-side encryption, see [#unique_62](#).

  - 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 amount of fees 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 header:value#header:value:...

```bash
./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.

```bash
./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 in the destination bucket or those have a later last modification time than that of the destination objects. The command is as follows:

```bash
./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 pay-by-requester 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 pay-by-requester enabled

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

• Copy a specified version of an object in a versioning-enabled bucket

```
./ossutil cp oss://bucket1/test.jpg oss://bucket2/ --version-id CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjIhYzQzZjNhYTAyZDE3MDRk
```

To use the `--version-id` option, you must run the `ls --all-versions` command to obtain version IDs of the object.

---

**Note:**

The `--version-id` option can only be used for objects in versioning-enabled buckets. For the command used to enable versioning on a bucket, see `bucket-versioning`.

• Copy only objects in the current directory and ignore subdirectories

```
./ossutil cp oss://bucket1/path oss://bucket2/path --only-current-dir -r
```

Performance optimization

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 operations when multiple objects 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.

By default, `ossutil` calculates the number of concurrent operations based on the object size. This option does not work for small objects. The object size threshold for large objects to be uploaded, downloaded, or copied in multipart mode can be
specified using the `--bigfile-threshold` option. When you upload, download, or copy multiple large 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 (such as network bandwidth, memory, and CPU) of your ECS instance or server are limited. If resources such as the network bandwidth, memory, and CPU are not fully occupied, you can increase the number of concurrent operations.
- If there are too many concurrent operations, inter-thread resource switching and snatching may degrade the upload, download, and copy performance of `ossutil` or lead to an EOF error. 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-r, --recursive</code></td>
<td>Recursively performs operations on objects in a bucket. If this option is specified, the command will perform operations on all objects in a bucket that meet the specified conditions. If this option is not specified, the command will only perform operations on a single specified object.</td>
</tr>
<tr>
<td><code>-f, --force</code></td>
<td>Forces an operation without prompting the user for confirmation.</td>
</tr>
<tr>
<td><code>-u, --update</code></td>
<td><code>ossutil</code> only uploads, downloads, or copies the objects that do not exist in the destination bucket or those have a later last modification time than that of the destination objects.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--output-dir</td>
<td>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.</td>
</tr>
<tr>
<td>--bigfile-threshold</td>
<td>Specifies the size of a large object for which to start resumable data transfer. Unit: bytes. Valid values: non-negative integers. Default value: 100 MB.</td>
</tr>
<tr>
<td>--part-size</td>
<td>Specifies the size of each part during multipart upload, download, or copy of large objects. Unit: bytes. By default, ossutil calculates the appropriate part size based on the object size. You can set this option to any positive integer if you need to optimize performance or are operating under special constraints.</td>
</tr>
<tr>
<td>--checkpoint-dir</td>
<td>Specifies the checkpoint directory. Default value: <code>.ossutil_checkpoint</code>. 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, make sure that the specified directory can be deleted.</td>
</tr>
<tr>
<td>--range</td>
<td>Specifies the byte range of the object to be downloaded. Bytes are numbered starting from 0.</td>
</tr>
<tr>
<td></td>
<td>• You can specify an interval. For example, a value of 3-9 indicates that the range is from byte 3 to byte 9 (including byte 3 and byte 9).</td>
</tr>
<tr>
<td></td>
<td>• You can specify the field from which the download starts. For example, 3- indicates that the range starts from byte 3 to the end of the object (including byte 3).</td>
</tr>
<tr>
<td></td>
<td>• You can specify the field with which the download ends. For example, -9 indicates that the range is from byte 0 to byte 9 (including byte 9).</td>
</tr>
<tr>
<td>--encoding-type</td>
<td>Specifies the method used to encode 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.</td>
</tr>
<tr>
<td>--include</td>
<td>Includes objects that match a specific string such as <code>*.jpg</code>.</td>
</tr>
<tr>
<td>--exclude</td>
<td>Excludes objects that match a specific string such as <code>*.txt</code>.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--meta</td>
<td>Configures the metadata of an object in [header:value#header:value...] format. Example: Cache-Control:no-cache#Content-Encoding:gz. For more information, see set-meta.</td>
</tr>
<tr>
<td>--acl</td>
<td>Configures the ACL for an object. Default value: default. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• default: inherited from the bucket</td>
</tr>
<tr>
<td></td>
<td>• private</td>
</tr>
<tr>
<td></td>
<td>• public-read</td>
</tr>
<tr>
<td></td>
<td>• public-read-write</td>
</tr>
</tbody>
</table>
## Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--snapshot-path</td>
<td>If you specify the --snapshot-path 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.</td>
</tr>
</tbody>
</table>

### Notice:
- The --snapshot-path 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 the 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.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--disable-crc64</code></td>
<td>Disables CRC-64. <code>ossutil</code> enables CRC-64 during data transmission by default.</td>
</tr>
<tr>
<td><code>--maxupspeed</code></td>
<td>Specifies the maximum upload speed in KB/s. The default value is 0, which indicates that there is no maximum upload speed.</td>
</tr>
<tr>
<td><code>--payer</code></td>
<td>Specifies the payer of the request. To enable pay-by-requester, set this option to requester.</td>
</tr>
<tr>
<td><code>--partition-download</code></td>
<td>Specifies the partition to download. The value of this option is in &quot;partition number:total number of partitions&quot; format. A value of 1:5 indicates that <code>ossutil</code> downloads partition 1 out of the total five partitions. Partitions are numbered starting from 1. Partitioning rules for objects are determined by <code>ossutil</code>. This option divides an object to be downloaded into multiple partitions that can be downloaded by multiple <code>ossutil</code> commands. Each <code>ossutil</code> command downloads its own partition. You can run multiple <code>ossutil</code> commands on different ECS instances in parallel.</td>
</tr>
<tr>
<td><code>-j, --jobs</code></td>
<td>Specifies the number of concurrent operations performed across multiple objects. Valid values: 1 to 10000. Default value: 3.</td>
</tr>
<tr>
<td><code>--parallel</code></td>
<td>Specifies the number of concurrent operations performed on a single object. Valid values: 1 to 10000. By default, the value of this option is determined by <code>ossutil</code> based on the operation type and object size.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>· info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>· debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td><code>--retry-times</code></td>
<td>Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.</td>
</tr>
<tr>
<td><code>--version-id</code></td>
<td>Specifies the version ID of an object to be downloaded or copied. The bucket in which the object is located must have versioning enabled.</td>
</tr>
<tr>
<td><code>--proxy-host</code></td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***.3128">http://120.79.***.3128</a> or socks5://120.79.<strong>.</strong>:1080.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username for the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password for the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--local-host</td>
<td>Specifies the local IP address of ossutil. If your computer has multiple IP addresses, you can specify this option so that ossutil can access OSS through the specified IP address.</td>
</tr>
<tr>
<td>--enable-symlink-dir</td>
<td>Specifies that the subdirectory to which the symbolic link points is uploaded. By default, subdirectories are not uploaded. The <code>probe</code> command can be used to check whether an object or a directory to which the symbolic link points is also a symbolic link.</td>
</tr>
<tr>
<td>--only-current-dir</td>
<td>Specifies that only objects in the current directory are uploaded, downloaded, or copied. The subdirectories in the current directory are ignored.</td>
</tr>
<tr>
<td>--disable-dir-object</td>
<td>Specifies that no OSS object is generated for the directory during object uploads.</td>
</tr>
<tr>
<td>--disable-all-symlink</td>
<td>All objects in the subdirectory to which the symbolic link points and the subdirectory to which the symbolic link points are ignored during object uploads.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#unique_69).

### 3.3.11 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_69](#unique_69).
Command syntax

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

Examples

```bash
./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 `stat` or `read-symlink` 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--encoding-type</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>--retry-times</code></td>
<td>Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| --loglevel | Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:  
|            | • info: generates prompt logs.  
|            | • debug: generates detailed logs that contain corresponding HTTP request and response information. |
| --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 [View all supported options](#).

### 3.3.12 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**

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

**Examples**

- **Query the capacity of a specify bucket**

  ```bash
  ./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**

  ```bash
  ./ossutil du oss://bucket1/test/
  object count:2    object sum size:64482
  part count:0     part sum size:0
  total du size(byte):64482
  ```
• View the size of a specified object

```bash
./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

```bash
./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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td><code>--proxy-host</code></td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.**:**1080.</td>
</tr>
<tr>
<td><code>--proxy-user</code></td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td><code>--proxy-pwd</code></td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
<tr>
<td><code>--payer</code></td>
<td>Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.</td>
</tr>
</tbody>
</table>

Note:
For more information about common options, see [View all supported options](#).
3.3.13 getallpartsize

The getallpartsize 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 getallpartsize 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 getallpartsize oss://bucket1
```

<table>
<thead>
<tr>
<th>PartNumber</th>
<th>UploadId</th>
<th>Size(Byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F18A92392DFD4B3FA897C267829FE417</td>
<td>52428800</td>
</tr>
<tr>
<td></td>
<td>oss://bucket1/test/test1.txt</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>F18A92392DFD4B3FA897C267829FE417</td>
<td>52428800</td>
</tr>
<tr>
<td></td>
<td>oss://bucket1/test/test1.txt</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>F18A92392DFD4B3FA897C267829FE417</td>
<td>52428800</td>
</tr>
<tr>
<td></td>
<td>oss://bucket1/test/test1.txt</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>F18A92392DFD4B3FA897C267829FE417</td>
<td>52428800</td>
</tr>
<tr>
<td></td>
<td>oss://bucket1/test/test1.txt</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>F18A92392DFD4B3FA897C267829FE417</td>
<td>52428800</td>
</tr>
<tr>
<td></td>
<td>oss://bucket1/test/test1.txt</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>F18A92392DFD4B3FA897C267829FE417</td>
<td>52428800</td>
</tr>
<tr>
<td></td>
<td>oss://bucket1/test/test1.txt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>total part count: 6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>total part size(MB): 300.00</td>
<td></td>
</tr>
</tbody>
</table>

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.<em>.</em>:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>
## Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**

For more information about common options, see [View all supported options](#).

### 3.3.14 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 `stat` 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

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

- Calculate the MD5 value of a local file

./ossutil hash test.txt --type=md5
MD5 : 01C3C45C03B2AF225EFAD9F911A33D73
Content-MD5 : AcPEXA0yryJe+tn5EaM9cw==

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--type</td>
<td>Specifies the algorithm that is used for calculation. Valid values: crc64 and md5. Default value: crc64.</td>
</tr>
</tbody>
</table>
| --loglevel | Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:  
  - info: generates prompt logs.  
  - debug: generates detailed logs that contain corresponding HTTP request and response information. |

Note:

For more information about common options, see View all supported options.
3.3.15 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
  ```

3.3.16 lifecycle

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

Note:
For more information about lifecycle rules, see #unique_70.

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
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>RuleID</ID>
    <Prefix>Prefix</Prefix>
    <Status>Status</Status>
    <Expiration>
      <Days>Days</Days>
    </Expiration>
    <Transition>
      <Days>Days</Days>
      <StorageClass>StorageClass</StorageClass>
    </Transition>
    <AbortMultipartUpload>
      <Days>Days</Days>
    </AbortMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

For more information about lifecycle configurations, see #unique_71.

- 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 lifecycle --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
<? xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <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
<? 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| --method   | Specifies the HTTP request method. Valid values:  
|            | • put: adds or modifies configurations  
|            | • get: obtains configurations  
|            | • delete: deletes configurations |
| --loglevel | Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:  
|            | • info: generates prompt logs.  
|            | • debug: generates detailed logs that contain HTTP requests and responses. |
| --proxy-host | Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 proxies 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 View all supported options.
3.3.17 listpart

The `listpart` command is used to list information of parts that are generated when multipart 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 15754AF7980C4DFB8193F190837520BB
```

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***.3128">http://120.79.***.3128</a> or socks5://120.79.**.*:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

Note:

For more information about common options, see [View all supported options](#).
3.3.18 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_72`.

Command syntax

- Add or modify logging configurations

  ```bash
  ./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

  ```bash
  ./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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--method</code></td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>- put: adds or modifies logging configurations.</td>
</tr>
<tr>
<td></td>
<td>- get: obtains logging configurations.</td>
</tr>
<tr>
<td></td>
<td>- delete: deletes logging configurations.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>- info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>- debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td><code>--proxy-host</code></td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.**.**:3128">http://120.79.**.**:3128</a> or socks5://120.79.<strong>.</strong>:1080.</td>
</tr>
<tr>
<td><code>--proxy-user</code></td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td><code>--proxy-pwd</code></td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**

For more information about common options, see [View all supported options](#).
3.3.19 ls

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

Command syntax

```
```

Examples

- **List buckets**

  ```
  ./ossutil ls
  CreationTime                              Region      StorageClass
  BucketName
  2016-10-21 16:18:37 +0800 CST             oss-cn-hangzhou  Archive
  oss://bucket1
  2016-12-01 15:06:21 +0800 CST             oss-cn-hangzhou  Standard
  oss://bucket2
  2016-07-20 10:36:24 +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 from which you want to start listing. The results displayed by ossutil are returned starting from the first bucket in alphabetical order after the value of the marker. In most cases, this parameter value is set to the last bucket name displayed on the previous page.

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

  ```
  ./ossutil ls oss:// --limited-num=1 -s
  oss://bucket1
  Bucket Number is:1
  ```
List all objects in a specified bucket

```
./ossutil ls oss:// --limited-num=1 --marker=bucket1
```

```
Bucket Number is:1
```

```
0.257636(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 on how to list the first two objects in the bucket1 root directory:

```
./ossutil ls oss://bucket1 --limited-num=1
```

```
Object Number is:1
```

```
0.007379(s) elapsed
```

$ ./ossutil ls oss://bucket1 --limited-num=1 --marker=a1

```
Object Number is:1
```

```
0.008392(s) elapsed
```

```
./ossutil ls oss://bucket1 --limited-num=1 --marker=a1
```

```
Object Number is:1
```

```
0.008392(s) elapsed
```
- List objects that meet specified conditions

You can set \(--\text{include}\) and \(--\text{exclude}\) options to list objects that meet specified conditions. For more information about \(--\text{include}\) and \(--\text{exclude}\), see \textit{cp}.

- List all objects that are not in \textit{JPG} format

\texttt{./ossutil ls oss://my-bucket/path \(--\text{exclude} \"*.jpg\"\)}

- List all objects that contain \textit{abc} in their names and are not in the \textit{JPG} and \textit{TXT} format.

\texttt{./ossutil ls oss://my-bucket1/path \(--\text{include} \"*abc*\" \(--\text{exclude} \"*.jpg\" \(--\text{exclude} \"*.txt\"\)}

- List upload IDs

You can add the \texttt{-m} option to the \texttt{ls} command to list upload IDs of all incomplete multipart upload tasks initiated to objects with a specified prefix.

\texttt{./ossutil ls oss://bucket1/obj1 \texttt{-m}}

<table>
<thead>
<tr>
<th>InitiatedTime</th>
<th>UploadID</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-01-13 03:45:26 +0000 CST</td>
<td>15754AF7980C4DFB8193F190837520BB</td>
</tr>
<tr>
<td>oss://bucket1/obj1</td>
<td></td>
</tr>
<tr>
<td>2017-01-13 03:43:13 +0000 CST</td>
<td>2A1F9B4A95E341BD9285CC42BB956E0</td>
</tr>
<tr>
<td>oss://bucket1/obj1</td>
<td></td>
</tr>
</tbody>
</table>

UploadID Number is: 2
0.070070(s) elapsed

- List all objects and incomplete multipart upload tasks

You can add the \texttt{-a} option to the \texttt{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.

\texttt{./ossutil ls oss://bucket1 \texttt{-a}}

<table>
<thead>
<tr>
<th>LastModifiedTime</th>
<th>Size(B)</th>
<th>StorageClass</th>
<th>ETAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-06-0514:06:29 +0000 CST</td>
<td>201933</td>
<td>Standard</td>
<td>7E2F4A7F1AC9D2F0996EB332D5EA5B41</td>
</tr>
<tr>
<td>2015-06-0514:36:21 +0000 CST</td>
<td>201933</td>
<td>Standard</td>
<td>6185CA2E8EB8510A61B3A845EAFE4174</td>
</tr>
<tr>
<td>2016-04-0814:50:47 +0000 CST</td>
<td>6476984</td>
<td>Standard</td>
<td>4F16FDAE7AC404CE8B727FCC67779D6</td>
</tr>
</tbody>
</table>

Object Number is: 3

InitiatedTime                     | UploadID                                      |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-01-1303:45:26 +0000 CST</td>
<td>15754AF7980C4DFB8193F190837520BB</td>
</tr>
<tr>
<td>oss://bucket1/obj1</td>
<td></td>
</tr>
<tr>
<td>2017-01-1303:43:13 +0000 CST</td>
<td>2A1F9B4A95E341BD9285CC42BB956E0</td>
</tr>
<tr>
<td>oss://bucket1/obj1</td>
<td></td>
</tr>
</tbody>
</table>
• **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**

You can use the `-d` option to list objects and subdirectories in the current directory, rather than recursively displaying all objects in all subdirectories.

```
./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 pay-by-requester mode**

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

• **List all versions of all objects in a versioning-enabled bucket**

```
./ossutil ls oss://bucket1 --all-versions
```

<table>
<thead>
<tr>
<th>LastModifiedTime</th>
<th>Size(B)</th>
<th>StorageClass</th>
<th>ETAG</th>
<th>VERSIONID</th>
<th>IS-LATEST</th>
<th>DELETE-MARKER</th>
<th>ObjectName</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-06-11 10:54:51 +0800 CST</td>
<td></td>
<td></td>
<td></td>
<td>CAEQARiBgICUsOuR2hYiIDI3NWVjNmEyYm0NTRk</td>
<td>true</td>
<td>true</td>
<td>oss://bucket1/test1.jpg</td>
</tr>
<tr>
<td>2019-06-11 11:03:37 +0800 CST</td>
<td></td>
<td></td>
<td></td>
<td>CAEQARiBgIDZtvuR2hYiIDNhYjRkN2M5NTA5OTl</td>
<td>true</td>
<td>true</td>
<td>oss://bucket1/test.jpg</td>
</tr>
<tr>
<td>2019-06-11 10:53:46 +0800 CST</td>
<td>118076</td>
<td>Standard</td>
<td>FFDB300F053AAF06F4C4C58A4869C427</td>
<td>CAEQARiBgID8rumR2hYiIGuy0TAyZGY2M5MjQ5</td>
<td>false</td>
<td>false</td>
<td>oss://bucket1/test1.jpg</td>
</tr>
<tr>
<td>2019-06-11 11:02:05 +0800 CST</td>
<td>345374</td>
<td>Standard</td>
<td>078A9852BCF81DC4581E6EDCBFD121BE</td>
<td>CAEQARiBgICNz_iR2hYiIGjJTBJbNDQxYnRnNTQ2</td>
<td>false</td>
<td>false</td>
<td>oss://bucket1/test.jpg</td>
</tr>
</tbody>
</table>
List all versions of a specific object in a versioning-enabled bucket

```
./ossutil ls oss://bucket1/test.jpg --all-versions
```

<table>
<thead>
<tr>
<th>LastModifiedTime</th>
<th>Size(B)</th>
<th>StorageClass</th>
<th>ETAG</th>
<th>VERSIONID</th>
<th>IS-LATEST</th>
<th>DELETE-MARKER</th>
<th>ObjectName</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-06-11 11:03:37 +0800 CST</td>
<td></td>
<td></td>
<td></td>
<td>CAEQARiBgIDZtvuR2hYiIDNhYjRkN2M5NTA5OT1l</td>
<td>true</td>
<td>true</td>
<td>oss://bucket1/test.jpg</td>
</tr>
<tr>
<td>2019-06-11 11:02:05 +0800 CST</td>
<td>345374</td>
<td>Standard</td>
<td>078A9852BCF81DC4811E6EDCBFD121BE</td>
<td>CAEQARiBgICNz_iR2hYiIGJjZTBjNDQxyWWrNTQ2</td>
<td>false</td>
<td>false</td>
<td>oss://bucket1/test.jpg</td>
</tr>
</tbody>
</table>

Object Number is: 2
0.361000(s) elapsed

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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-s</code></td>
<td>Lists items in short format.</td>
</tr>
<tr>
<td><code>-d</code></td>
<td>Lists objects and subdirectories in the current directory, rather than recursively displaying all objects in all subdirectories.</td>
</tr>
<tr>
<td><code>-m</code></td>
<td>Lists only incomplete multipart upload tasks in the bucket. Objects are not displayed.</td>
</tr>
<tr>
<td><code>-a</code></td>
<td>Lists both objects and incomplete multipart upload tasks in the bucket.</td>
</tr>
<tr>
<td><code>--limited-num</code></td>
<td>Specifies the maximum number of returned entries. This option can be used together with the <code>--marker</code> option to list objects by page.</td>
</tr>
<tr>
<td><code>--marker</code></td>
<td>Specifies the bucket name, object name, or multipart upload ID from which you want to start listing.</td>
</tr>
<tr>
<td><code>--upload-id-marker</code></td>
<td>Specifies the multipart upload ID from which you want to start listing.</td>
</tr>
<tr>
<td><code>--payer</code></td>
<td>Specifies the payer of the request. If pay-by-requester is enabled, you can set this option to requester.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>· info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>· debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--include</td>
<td>Includes objects that match a specified string, such as *.jpg.</td>
</tr>
<tr>
<td>--exclude</td>
<td>Excludes objects that match a specified string, such as *.txt.</td>
</tr>
<tr>
<td>--all-versions</td>
<td>Lists all versions of a specified object. This option can only be used for objects in a bucket that has versioning enabled.</td>
</tr>
<tr>
<td>--version-id-marker</td>
<td>Specifies the object version from which you want to start listing.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***.***:3128">http://120.79.***.***:3128</a> or socks5://120.79.<em><strong>.</strong></em>:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

Note:
For more information about common options, see View all supported options.
3.3.20 mb

The mb command is used to create a bucket.

Command syntax

```bash
./ossutil mb oss://bucketname [--acl=ACL][--storage-class sc][--redundancy-type type]
```

Examples

- Create a bucket
  ```bash
  ./ossutil mb oss://bucket1
  ```

  **Notice:**
  The bucket name must be unique. If a bucket of the specified name already exists, an error message is displayed.

- Specify the access control list (ACL) when creating a bucket
  ```bash
  ./ossutil mb oss://bucket1 --acl public-read-write
  ```
  
  **--acl:** Specifies the ACL for the bucket. Default value: private.
  - private
  - public-read
  - public-read-write
  
  For more information about ACLs, see #unique_15.

- Specify the storage class when creating a bucket
  ```bash
  ./ossutil mb oss://bucket1 --storage-class IA
  ```
  
  **--storage-class:** specifies the default storage class for the bucket. Default value: Standard.
  - Standard
  - IA
  - Archive
  
  For more information about storage classes, see #unique_16.
• Create a bucket in a specified region

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

• Create a bucket from a specified configuration file

```bash
./ossutil mb oss://bucket1 -c your_config_file_path
```

• Create a bucket and enable zone-redundant storage

```bash
./ossutil mb oss://bucket1 --redundancy-type ZRS
```

`--redundancy-type`: specifies the data redundancy type for the bucket. Valid values: LRS (locally redundant storage) and ZRS (zone-redundant storage). Default value: LRS.

**Note:**
Zone-redundant storage is available only in China (Shenzhen), China (Beijing), and China (Shanghai) regions. If you set this parameter to ZRS when creating a bucket, you must configure an endpoint corresponding to the region. For more information about zone-redundant storage, see #unique_74.

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--acl</code></td>
<td>Specifies the ACL for the bucket. Default value: private. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• private</td>
</tr>
<tr>
<td></td>
<td>• public-read</td>
</tr>
<tr>
<td></td>
<td>• public-read-write</td>
</tr>
<tr>
<td></td>
<td>For more information about ACLs, see #unique_15.</td>
</tr>
<tr>
<td><code>--storage-class</code></td>
<td>Specifies the default storage class for the bucket. Default value:</td>
</tr>
<tr>
<td></td>
<td>Standard. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• Standard</td>
</tr>
<tr>
<td></td>
<td>• IA</td>
</tr>
<tr>
<td></td>
<td>• Archive</td>
</tr>
<tr>
<td></td>
<td>For more information about storage classes, see #unique_16.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-c, --config-file</td>
<td>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 <code>config</code>.</td>
</tr>
<tr>
<td>-e, --endpoint</td>
<td>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.</td>
</tr>
<tr>
<td>-L, --language</td>
<td>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.</td>
</tr>
</tbody>
</table>
| --loglevel              | Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:  
  - 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.                                                                          |
| --redundancy-type       | Specifies the data redundancy type of a bucket. Valid values: LRS (locally redundant storage) and ZRS (zone-redundant storage). Default value: LRS. |

**Note:**

For more information about common options, see [View all supported options](#).
3.3.21 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--encoding-type</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>- info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>- debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
</tbody>
</table>
### 3.3.22 object-tagging

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

#### 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 a 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

  
  ```
  ```

- Delete object tagging configurations

  
  ```
  ```

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
  ```
Add tags to a specified version of an object in a versioning-enabled bucket

```
./ossutil object-tagging --method put oss://bucket1/test.png A#B
   --version-id CAEQCBiBgMCwrPXx8xYiIDA1NDVkZDI3ZDI5Nzg5MWE0NjNiNzg80WE5ZGNh
```

To use the `--version-id` option, you must run the `ls --all-versions` command to obtain all available versions of the object.

---

**Note:**
The `--version-id` option can only be used for objects in versioning-enabled buckets. For more information about the command used to enable versioning on a bucket, see `bucket-versioning`.

---

**Common options**

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--method</code></td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• put: adds or modifies object tagging configurations.</td>
</tr>
<tr>
<td></td>
<td>• get: obtains object tagging configurations.</td>
</tr>
<tr>
<td></td>
<td>• delete: deletes object tagging configurations.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that</td>
</tr>
<tr>
<td></td>
<td>no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding</td>
</tr>
<tr>
<td></td>
<td>HTTP request and response information.</td>
</tr>
<tr>
<td><code>--proxy-host</code></td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported.</td>
</tr>
<tr>
<td></td>
<td>An example of the URL is <a href="http://120.79.***.3128">http://120.79.***.3128</a> or socks5://120.79.***.1080.</td>
</tr>
<tr>
<td><code>--proxy-user</code></td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td><code>--proxy-pwd</code></td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
<tr>
<td><code>--payer</code></td>
<td>Specifies the payer of the request. To enable pay-by-requester, set this</td>
</tr>
<tr>
<td></td>
<td>option to requester.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-j, --jobs</td>
<td>Specifies the number of concurrent operations performed across multiple objects. Valid values: 1 to 10000. Default value: 3.</td>
</tr>
<tr>
<td>-r, --recursive</td>
<td>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.</td>
</tr>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--version-id</td>
<td>Specifies the version ID of an object to be downloaded or copied. The bucket in which the object is located must have versioning enabled.</td>
</tr>
</tbody>
</table>

**Note:**

For more information about common options, see View all supported options.
3.3.23 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 by 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 by 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=TMP.xxxxxxxx5r9f1FV12y8_Qis6LUVmvoSCUSs7aboCCHtydQ0axN32Sn-UvyY3AAAaWAIUarYNLc087AKMEcE503AxxxxxxoCFAQuRdZYyVFyqOW8QkGAN-bamU1Q&Signature=bIa4llbMbltlr17rwcK%2FXXvTtxw%3D".

Note:
For more 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**: 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 an 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 the **file_name** parameter 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 a 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

    ```bash
    ./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

    ```bash
    ./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 a 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

**Other probe**

```
./ossutil probe --probe-item item_value --bucketname bucket-name [--object=object_name]
```

**--probe-item**: specifies the content for other probe. Valid values: upload-speed, download-speed, and cycle-symlink.

- **upload-speed**: checks the upload bandwidth and generates the troubleshooting results. If this parameter is specified, **--object=object_name** is optional.

- **download-speed**: checks the download bandwidth and generates the troubleshooting results. You must specify **--object=object_name**, and the specified object must exist in OSS. To ensure the accuracy, the size of the specified object must exceed 10 MB.

- **cycle-symlink**: checks whether a symbolic link in the local file directory also points to a symbolic link that points to itself.

**Examples**

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

  ```
  ```
• Download an object from a bucket and generate a troubleshooting report

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

• Check the upload result and generate a troubleshooting report

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

• Check the upload bandwidth and generate the troubleshooting results

```
./ossutil probe --probe-item upload-speed --bucketname bucket1
cpu core count:4
parallel:4,average speed:14231.17(KB/s),current speed:12992.00(KB/s)
,max speed:23776.00(KB/s)
parallel:5,average speed:16871.72(KB/s),current speed:14272.00(KB/s)
, max speed:22336.00(KB/s)
parallel:6,average speed:21072.55(KB/s),current speed:22176.00(KB/s)
, max speed:25344.00(KB/s)
parallel:7,average speed:24622.34(KB/s),current speed:25216.00(KB/s)
, max speed:31200.00(KB/s)
parallel:8,average speed:26388.97(KB/s),current speed:30880.00(KB/s)
, max speed:30880.00(KB/s)
suggest parallel is 8, max average speed is 26388.97(KB/s)
152.834561(s) elapsed
```

• Checks the download bandwidth and generate the troubleshooting results

```
./ossutil probe --probe-item download-speed --bucketname bucket1 --object test/1.bmp
cpu core count:4
parallel:4,average speed:25925.10(KB/s),current speed:26823.95(KB/s)
, max speed:30149.93(KB/s)
parallel:5,average speed:31565.45(KB/s),current speed:30124.70(KB/s)
, max speed:37400.07(KB/s)
parallel:6,average speed:34793.28(KB/s),current speed:46594.91(KB/s)
, max speed:46594.91(KB/s)
parallel:7,average speed:40403.66(KB/s),current speed:46594.91(KB/s)
, max speed:54165.61(KB/s)
suggest parallel is 8, max average speed is 48459.10(KB/s)
152.724271(s) elapsed
```

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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--url</td>
<td>Specifies the network address of the object. ossutil will use this address to download the object.</td>
</tr>
<tr>
<td>--bucketname</td>
<td>Specifies the name of the bucket in OSS.</td>
</tr>
<tr>
<td>--object</td>
<td>Specifies the name of the object in OSS.</td>
</tr>
<tr>
<td>--addr</td>
<td>Specifies the domain name to ping. Default value: <a href="http://www.aliyun.com">www.aliyun.com</a>.</td>
</tr>
<tr>
<td>--upmode</td>
<td>Specifies the upload method. Default value: normal. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• normal</td>
</tr>
<tr>
<td></td>
<td>• append</td>
</tr>
<tr>
<td></td>
<td>• multipart</td>
</tr>
<tr>
<td>--upload</td>
<td>Uploads the object to OSS.</td>
</tr>
<tr>
<td>--download</td>
<td>Downloads the object from OSS.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.***:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--probe-item</td>
<td>Specifies the items to be checked by using the probe command. Valid values: upload-speed, download-speed, cycle-symlink: checks whether a symbolic link in the local file directory also points to a symbolic link that points to itself.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see View all supported options.

### 3.3.24 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_69.
- For more information about the command used to create symbolic links, see `create-symlink`.

**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.

```bash
./ossutil read-symlink oss://bucket1/object1
Etag : 455E20DBFFF1D58BBB7D092C46B16DB6
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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--encoding-type</code></td>
<td>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.</td>
</tr>
</tbody>
</table>
| `--loglevel`    | Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:  
|                 | • 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 `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 [View all supported options](https://www.example.com).
3.3.25 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_77.

Command syntax

- **Add or modify hotlink protection configurations for a specified bucket**

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

  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-referrer**: 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**

  ```bash
  ./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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--method</code></td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• put: adds or modifies hotlink protection configurations.</td>
</tr>
<tr>
<td></td>
<td>• get: obtains hotlink protection configurations.</td>
</tr>
<tr>
<td></td>
<td>• delete: deletes hotlink protection configurations.</td>
</tr>
<tr>
<td><code>--disable-empty-referer</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td><code>--proxy-host</code></td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***.3128">http://120.79.***.3128</a> or socks5://120.79.**.*.1080.</td>
</tr>
<tr>
<td><code>--proxy-user</code></td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td><code>--proxy-pwd</code></td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>
3.3.26 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
  ```bash
  ./ossutil restore oss://bucket/object
  ```

- Restore all objects that have a specified prefix to the readable state
  ```bash
  ./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.

- Restore a specified version of an object in a versioning-enabled bucket to the readable state
  ```bash
  ./ossutil restore oss://bucket1/test.jpg --version-id CAEQARiBgIGuIGyOTAyZGY2MzU5MjQ5ZjlhYzQzNjYlYTAyZDE3MDRk
  ```

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

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r, --recursive</td>
<td>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.</td>
</tr>
<tr>
<td>-f, --force</td>
<td>Forces an operation without prompting the user for confirmation.</td>
</tr>
<tr>
<td>-j, --jobs</td>
<td>Specifies the number of concurrent operations performed across multiple objects. Valid values: 1 to 10000. Default value: 3.</td>
</tr>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--output-dir=</td>
<td>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.</td>
</tr>
<tr>
<td>--version-id</td>
<td>Specifies the version ID of an object. The bucket in which the object is located must have versioning enabled.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>· info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>· debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--retry-times</td>
<td>Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--payer</td>
<td>Specifies the payer of the request. To enable pay-by-requester, set this option to requester.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***.3128">http://120.79.***.3128</a> or socks5://120.79.***.1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#).

### 3.3.27 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_14](#).

#### Command syntax

- Set the pay-by-requester mode

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

  **Valid values of** `payment_parameter` **are Requester and BucketOwner.**

  - **Requester:** enables the pay-by-requester mode. Requesters pay the cost of requests and the data download from buckets.
  - **BucketOwner:** 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 BucketOwner

• Query the pay-by-requester configurations

./ossutil request-payment --method get oss://bucket1 BucketOwner
0.178036(s) elapsed

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--method</td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• put: enables or disables the pay-by-requester mode.</td>
</tr>
<tr>
<td></td>
<td>• get: obtains pay-by-requester configurations.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log</td>
</tr>
<tr>
<td></td>
<td>files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request</td>
</tr>
<tr>
<td></td>
<td>and response information.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are</td>
</tr>
<tr>
<td></td>
<td>supported. An example of the URL is <a href="http://120.79.***.3128">http://120.79.***.3128</a> or socks5://120.</td>
</tr>
<tr>
<td></td>
<td>79.<strong>.</strong>:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

Note:
3.3.28 rm

The `rm` command is used to delete a bucket, an object, or a 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 configure 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**

  ```bash
  ./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.

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

- **Delete multiple objects that meet specified conditions**

  You can configure `--include` and `--exclude` options to delete objects that meet specified conditions. For more information about `--include` and `--exclude`, see `cp`.

  - Delete all objects except `.jpg` objects.

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

  - Delete all objects that contain `abc` in their names and are not in `JPG` or `TXT` format.

    ```bash
    ./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.

  ```bash
  ./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.

  ```bash
  ./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.
  ```
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.
```

Delete a specified version of an object in a versioning-enabled bucket

```
./ossutil.exe rm oss://bucket1/test.jpg --version-id CAEQARiBgI D8rumR2hYiIGUyOTAyZGY2MzU5MjQSZjlhYzQzJnYlYTAyZDE3MDRk
```

To use the `--version-id` option, you must run the `ls --all-versions` command to obtain version IDs of the object.

**Note:**
The `--version-id` option can only be used for objects in versioning-enabled buckets. For more information about the command used to enable versioning on a bucket, see `bucket-versioning`.

Delete all versions of a specified object in a versioning-enabled bucket

```
./ossutil.exe rm oss://bucket1/test.jpg --all-versions
Succeed: Total 2 objects. Removed 2 objects.
```

Delete all versions of all objects in a versioning-enabled bucket

```
./ossutil.exe rm oss://bucket1/ --all-versions -r
```

Common options

The following table describes the options you can add to the `rm` command to delete different content.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r, --recursive</td>
<td>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.</td>
</tr>
<tr>
<td>-b, --bucket</td>
<td>Specifies the bucket on which to perform an operation.</td>
</tr>
<tr>
<td>-m, --multipart</td>
<td>Specifies that operations are only to be performed on uncompleted multipart upload tasks in a bucket instead of objects.</td>
</tr>
<tr>
<td>-a, --all-type</td>
<td>Specifies that operations are to be performed on both the objects and uncompleted multipart upload tasks in a bucket.</td>
</tr>
<tr>
<td>-f, --force</td>
<td>Forces an operation without prompting the user for confirmation.</td>
</tr>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--retry-times=</td>
<td>Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>- info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>- debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--include</td>
<td>Includes objects that match a specified string, such as *.jpg.</td>
</tr>
<tr>
<td>--exclude</td>
<td>Excludes objects that match a specified string, such as *.txt.</td>
</tr>
<tr>
<td>--version-id</td>
<td>Deletes a specified version of an object. This option can only be used for objects in a versioning-enabled bucket.</td>
</tr>
<tr>
<td>--all-versions</td>
<td>Deletes all versions of a specified object. This option can only be used for objects in a versioning-enabled bucket.</td>
</tr>
<tr>
<td>--payer</td>
<td>Specifies the payer of the request. To enable pay-by-requester, set this option to requester.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.***:1080.</td>
</tr>
</tbody>
</table>
### Object Storage Service

#### Tools / ossutil

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#).

#### 3.3.29 set-acl

set-acl is used to configure the access control list (ACL) for a bucket or an 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 ACLs:
  - 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 ACLs:
  - default: The object inherits the ACL of the bucket to which it belongs.
  - 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 `--include` and `--exclude` to select objects that meet specified conditions. For more information, see `cp`.

    ```
    - Set the ACL to private for all objects except objects in JPG format.
      ```
      
      ```
      ./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 JPG or TXT format

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

  - Configure the ACL for a specified version of an object in a versioning-enabled bucket

    ```
    ./ossutil set-acl oss://bucket1/test.jpg private --version-id CAEQARiBgID8rumR2hYiIGUyOTAYzGY2MzU5MjQ5ZjIhYzQzZjNlTAYzDE3MDrk
    ```

    To use the `--version-id` option, you must run the `ls --all-versions` command to obtain version IDs of the object.

    **Note:**

    The `--version-id` option can only be used for objects in versioning-enabled buckets. For more information about the command used to enable versioning on a bucket, see `bucket-versioning`.

**Common options**

The following table describes the options you can add to the `set-acl` command to set different ACLs for different objects.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r, --recursive</td>
<td>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.</td>
</tr>
<tr>
<td>-b, --bucket</td>
<td>Specifies the bucket on which to perform an operation.</td>
</tr>
<tr>
<td>-f, --force</td>
<td>Forces an operation without prompting the user for confirmation.</td>
</tr>
<tr>
<td>--include</td>
<td>Includes objects that match a specified string, such as *.jpg.</td>
</tr>
<tr>
<td>--exclude</td>
<td>Excludes objects that match a specified string, such as *.txt.</td>
</tr>
<tr>
<td>-j, --jobs</td>
<td>Specifies the number of concurrent tasks when multiple objects are operated. Valid values: 1 to 10000. Default value: 3.</td>
</tr>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--output-dir</td>
<td>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.</td>
</tr>
</tbody>
</table>
| --loglevel       | Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:  
|                  |   • info: generates prompt logs.  
<p>|                  |   • 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.                          |
| --version-id     | Specifies the version ID of an object in a versioning-enabled bucket.                                                                       |
| --proxy-host     | Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.*.*.*:3128">http://120.79.*.*.*:3128</a> or socks5://120.79.<em>.</em>.*:1080. |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see View all supported options.

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

**Command syntax**

```
./ossutil set-meta oss://bucket[/prefix] [header:value#header:value ...
```

**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-Disposition
- Accept-Encoding
- X-Oss-Server-Side-Encryption
Content-Type
Headers with the X-Oss-Meta- prefix

For more information, see #unique_78.

- 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 `--include` and `--exclude` 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 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
```

- Configure the metadata of a specified version of an object in a versioning-enabled bucket

```
./ossutil set-meta oss://bucket1/test.jpg X-Oss-Storage-Class:Standard
--version-id CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5
ZjlhYzQzZjNlYTAyZDE3MDRk
```

To use the `--version-id` option, you must run the `ls --all-versions` command to obtain all available versions of the object.

**Note:**
The `--version-id` option can only be used for objects in versioning-enabled buckets. For more information about the command used to enable versioning on a bucket, see `bucket-versioning`.

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-u, --update</code></td>
<td>Updates the metadata of a specified object.</td>
</tr>
<tr>
<td><code>--delete</code></td>
<td>Deletes the metadata of a specified object.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-r, --recursive</td>
<td>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.</td>
</tr>
<tr>
<td>-f, --force</td>
<td>Forces an operation without prompting the user for confirmation.</td>
</tr>
<tr>
<td>--include</td>
<td>Includes objects that match a specified string, such as *.jpg.</td>
</tr>
<tr>
<td>--exclude</td>
<td>Excludes objects that match a specified string, such as *.txt.</td>
</tr>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>-j, --jobs</td>
<td>Specifies the number of concurrent tasks when multiple objects are operated. Valid values: 1 to 10000. Default value: 3.</td>
</tr>
<tr>
<td>-L, --language</td>
<td>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.</td>
</tr>
<tr>
<td>--output-dir</td>
<td>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.</td>
</tr>
<tr>
<td>--logfile</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values: info: generates prompt logs. debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--version-id</td>
<td>Specifies the version ID of an object in a versioning-enabled bucket.</td>
</tr>
<tr>
<td>--retry-times</td>
<td>Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.***:1080.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

Note:
For more information about common options, see View all supported options.

3.3.31 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 get the link address of the object in OSS.

Command syntax

```
./ossutil sign oss://bucket/object [--timeout t] [--traffic-limit limitSpeed] [--disable-encode-slash]
```

- **--timeout**: specifies the timeout value of the object URL. Unit: seconds. Valid values: non-negative integers. Default value: 60.
- **--traffic-limit**: specifies the speed for accessing the object over HTTP. Unit: bit/s. Valid values: 819200 to 838860800 (100 KB/s to 100 MB/s). Default value: 0 (unlimited).

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 access speed of 1 MB/s
  ```
  ./ossutil sign oss://bucket/path/object --traffic-limit 8388608
  ```

Note:
Object Storage Service

Tools / 3 ossutil

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

- Generate the URL for a specified version of an object in a bucket that has
  versioning enabled

  ./ossutil sign oss://bucket1/test.jpg --timeout 3600 --version-id
  CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5ZjIhYzQzZjNlYTAyZDE3M9Drk

  To use the --version-id option, you must run the ls --all-versions command to obtain
  all available versions of the object.

  Note:
The --version-id option can only be used for objects in versioning-enabled
  buckets. For more information about the command used to enable versioning on
  a bucket, see bucket-versioning.

Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--timeout</td>
<td>Specifies the timeout value of the object URL. Unit: seconds. Valid values: non-negative integers. Default value: 60.</td>
</tr>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--version-id</td>
<td>Specifies the version ID of an object in a versioning-enabled bucket.</td>
</tr>
<tr>
<td>--trafic-limit</td>
<td>Specifies the speed for accessing the object over HTTP. Unit: bit/s. Valid values: 819200 to 838860800 (100 KB/s to 100 MB/s). Default value: 0 (unlimited).</td>
</tr>
</tbody>
</table>
| --disable-
  encode-slash  | Specifies that forward slashes (/) in the URL path are not encoded. |
3.3.32 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--encoding-type</td>
<td>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.</td>
</tr>
</tbody>
</table>

Note:

For more information about common options, see View all supported options.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--retry-times</td>
<td>Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.</td>
</tr>
<tr>
<td>--payer</td>
<td>Specifies the payer of the request. To enable the pay-by-requester mode, set this option to requester.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.***:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

**Note:**
For more information about common options, see [View all supported options](#).

### 3.3.33 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
```
Common options

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f</code>, <code>--force</code></td>
<td>Forces an operation without prompting the user for confirmation.</td>
</tr>
<tr>
<td><code>--retry-times</code></td>
<td>Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.</td>
</tr>
<tr>
<td><code>--loglevel</code></td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• <code>info</code>: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• <code>debug</code>: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td><code>--proxy-host</code></td>
<td>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>.</td>
</tr>
<tr>
<td><code>--proxy-user</code></td>
<td>Specifies the username of the proxy server. The default value is null.</td>
</tr>
<tr>
<td><code>--proxy-pwd</code></td>
<td>Specifies the password of the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

Note:
For more information about common options, see `View all supported options`.

3.3.34 website

The `website` command is used to add, modify, query, or delete static website hosting, redirection-based back-to-origin, and mirroring-based back-to-origin configurations of a bucket.

Note:
- For more information about static website hosting, see `#unique_79`.
- For more information about back-to-origin, see `#unique_80`.
Command syntax

- Add or modify website-related configurations

```bash
./ossutil website --method put oss://bucket local_xml_file
```

ossutil is used to read website-related configurations from the `local_xml_file` configuration file and add the rules obtained from the configuration file for the bucket. If the bucket already has website-related configurations, the new configurations will overwrite the existing configurations.

**Note:**

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

- The `IndexDocument` field specifies the default homepage.
- The `ErrorDocument` field specifies the default 404 page.
- The `RoutingRules` field configures redirection- and mirroring-based back-to-origin rules. For more information, see Configure back-to-origin rules.

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

```xml
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <RuleNumber>1</RuleNumber>
      <Condition>
        <KeyPrefixEquals>abc/</KeyPrefixEquals>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <RedirectType>Mirror</RedirectType>
        <PassQueryString>true</PassQueryString>
        <MirrorURL>http://www.test.com/</MirrorURL>
        <MirrorPassQueryString>true</MirrorPassQueryString>
        <MirrorFollowRedirect>true</MirrorFollowRedirect>
        <MirrorCheckMd5>false</MirrorCheckMd5>
        <MirrorHeaders>
          <PassAll>true</PassAll>
          <Pass>myheader-key1</Pass>
          <Pass>myheader-key2</Pass>
          <Remove>myheader-key3</Remove>
          <Remove>myheader-key4</Remove>
        </MirrorHeaders>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```
<Set>
  <Key>myheader-key5</Key>
  <Value>myheader-value5</Value>
</Set>
</Redirect>
</RoutingRule>
<RoutingRule>
  <Condition>
    <KeyPrefixEquals>abc/</KeyPrefixEquals>
    <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
  </Condition>
  <IncludeHeader>
    <Key>host</Key>
    <Equals>test.oss-cn-beijing-internal.aliyuncs.com</Equals>
  </IncludeHeader>
  <Redirect>
    <RedirectType>AliCDN</RedirectType>
    <Protocol>http</Protocol>
    <HostName>www.test.com</HostName>
    <PassQueryString>false</PassQueryString>
    <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith>
    <HttpRedirectCode>301</HttpRedirectCode>
  </Redirect>
</RoutingRule>
</RoutingRules>
</WebsiteConfiguration>

• 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 are saved as a local file. If this parameter is not specified, ossutil displays 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"?>
<WebsiteConfiguration>
```
Add back-to-origin configurations

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

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

```xml
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
  <RoutingRules>
    <RoutingRule>
      <RuleNumber>1</RuleNumber>
      <Condition>
        <KeyPrefixEquals>abc/</KeyPrefixEquals>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <RedirectType>Mirror</RedirectType>
        <PassQueryString>true</PassQueryString>
        <MirrorURL>http://www.aliyun.com/</MirrorURL>
        <MirrorPassQueryString>true</MirrorPassQueryString>
        <MirrorFollowRedirect>true</MirrorFollowRedirect>
        <MirrorCheckMd5>false</MirrorCheckMd5>
        <MirrorHeaders>
          <PassAll>true</PassAll>
          <Pass>myheader1</Pass>
          <Pass>myheader2</Pass>
          <Remove>myheader3</Remove>
          <Remove>myheader4</Remove>
          <Set>
            <Key>myheader5</Key>
            <Value>myheader5</Value>
          </Set>
        </MirrorHeaders>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```

Obtain website-related configurations

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

Delete website-related configurations

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

Common options

The following table describes the options you can add to the `website` command.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--method</td>
<td>Specifies the HTTP request method. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• put: adds or modifies configurations.</td>
</tr>
<tr>
<td></td>
<td>• get: obtains configurations.</td>
</tr>
<tr>
<td></td>
<td>• delete: deletes configurations.</td>
</tr>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated.</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain HTTP request and response information.</td>
</tr>
<tr>
<td>--proxy-host</td>
<td>Specifies the URL of the proxy server. HTTP, HTTPS, and SOCKS5 are supported. An example of the</td>
</tr>
<tr>
<td></td>
<td>URL is <a href="http://120.79.***:3128">http://120.79.***:3128</a> or socks5://120.79.***:1080.</td>
</tr>
<tr>
<td>--proxy-user</td>
<td>Specifies the username for the proxy server. The default value is null.</td>
</tr>
<tr>
<td>--proxy-pwd</td>
<td>Specifies the password for the proxy server. The default value is null.</td>
</tr>
</tbody>
</table>

Note:
For more information about common options, see [View all supported options](#).

### 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 manage buckets across regions, 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 IDs.

-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 secrets.
- **--loglevel**

  Generates the ossutil log file `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.

      ```bash
      ./ossutil [command] --loglevel=info
      ```
    - **debug:** generates logs that contain HTTP requests and responses and original signature strings to locate problems.

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

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

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

  **Example:**

  ```bash
  ./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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s, --short-format</td>
<td>Lists items in short format. The long format is used if this option is not specified.</td>
</tr>
<tr>
<td>--bigfile-threshold</td>
<td>Specifies the size of a large object for which to start resumable data transfer. Unit: bytes. Valid values: non-negative integers. Default value: 100 MB.</td>
</tr>
<tr>
<td>--acl</td>
<td>Configures the access control list (ACL) for an object or a bucket.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--range</td>
<td>Specifies the byte range of the object to be downloaded. Bytes are numbered from 0.</td>
</tr>
<tr>
<td></td>
<td>• You can specify a range. For example, 3–9 indicates that the range is from byte 3 to byte 9 (including byte 3 and byte 9).</td>
</tr>
<tr>
<td></td>
<td>• You can specify the field from which the download starts. For example, 3– indicates that the range is from byte 3 to the end of the object (including byte 3).</td>
</tr>
<tr>
<td></td>
<td>• You can specify the field with which the download ends. For example, –9 indicates that the range is from byte 0 to byte 9 (including byte 9).</td>
</tr>
<tr>
<td>--all-versions</td>
<td>Specifies all versions of an object.</td>
</tr>
<tr>
<td>--type</td>
<td>Specifies the algorithm that is used for calculation. Valid values: crc64 and md5. Default value: crc64.</td>
</tr>
<tr>
<td>-v, --version</td>
<td>Displays the ossutil version and exits.</td>
</tr>
<tr>
<td>-u, --update</td>
<td>Updates the ossutil version.</td>
</tr>
<tr>
<td>--origin</td>
<td>Specifies the value of the Origin header in an HTTP request.</td>
</tr>
<tr>
<td>--upmode</td>
<td>Specifies the upload method. Valid values: normal, append, and multipart. Default value: normal. This option is used with the probe command.</td>
</tr>
<tr>
<td>--sse-algorithm</td>
<td>Specifies the server-side encryption algorithm. Valid values: KMS and AES256.</td>
</tr>
<tr>
<td>--include</td>
<td>Includes objects that match a specific string such as *.jpg.</td>
</tr>
<tr>
<td>--exclude</td>
<td>Excludes objects that match a specific string such as *.txt.</td>
</tr>
<tr>
<td>-r, --recursive</td>
<td>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.</td>
</tr>
<tr>
<td>--addr</td>
<td>Specifies a network address, which is a domain name in most cases. This option is used with the probe command.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--kms-masterkey-id</td>
<td>Specifies the CMK ID used for encryption in Key Management Service (KMS).</td>
</tr>
<tr>
<td>--version-id</td>
<td>Specifies the version ID of an object.</td>
</tr>
<tr>
<td>--version-id-marker</td>
<td>Specifies the object version from which you want to start listing.</td>
</tr>
<tr>
<td>-m, --multipart</td>
<td>Specifies that operations are only to be performed on uncompleted multipart upload tasks in a bucket instead of objects.</td>
</tr>
<tr>
<td>-d, --directory</td>
<td>Lists objects and subdirectories in the current directory, instead of recursively displaying all objects in all subdirectories.</td>
</tr>
<tr>
<td>--payer</td>
<td>Specifies the payer of the request. If pay-by-requester is enabled, you can set this option to requester.</td>
</tr>
<tr>
<td>--maxupspeed</td>
<td>Specifies the maximum upload speed. Unit: KB/s. Default value: 0 (unlimited).</td>
</tr>
<tr>
<td>--retry-times</td>
<td>Specifies the number of times an operation is retried if the operation fails. Valid values: 1 to 500. Default value: 10.</td>
</tr>
<tr>
<td>-c, --config-file</td>
<td>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.</td>
</tr>
<tr>
<td>--download</td>
<td>Downloads an object from OSS. This option is used with the probe command.</td>
</tr>
<tr>
<td>-j, --jobs</td>
<td>Specifies the number of concurrent operations performed across multiple objects. Valid values: 1 to 10000. Default value: 3.</td>
</tr>
<tr>
<td>-a, --all-type</td>
<td>Specifies that operations are to be performed on both the objects and uncompleted multipart upload tasks in a bucket.</td>
</tr>
<tr>
<td>--disable-empty-referer</td>
<td>Indicates that the referer field cannot be left empty. This option is used with the referer command.</td>
</tr>
<tr>
<td>--method</td>
<td>Specifies the HTTP request method, which can be PUT, GET, or DELETE.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--output-dir</td>
<td>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 report 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.</td>
</tr>
<tr>
<td>--meta</td>
<td>Configures the metadata of an object in <code>[header:value#header:value...]</code> format. Example: <code>Cache-Control: no-cache#Content-Encoding: gzip</code>.</td>
</tr>
<tr>
<td>--object</td>
<td>Specifies the name of an object in OSS. This option is used with the <code>probe</code> command.</td>
</tr>
<tr>
<td>-e, --endpoint</td>
<td>Specifies the basic endpoint of ossutil. This option value will overwrite the corresponding endpoint in the configuration file. For more information about region endpoints, see <code>#unique_11</code>.</td>
</tr>
<tr>
<td>--limited-num</td>
<td>Specifies the maximum number of returned results.</td>
</tr>
<tr>
<td>-L, --language</td>
<td>Specifies the language ossutil uses. Valid values: CH and EN. Default value: CH. To set this option to CH, make sure that your system supports UTF-8 encoding.</td>
</tr>
<tr>
<td>--delete</td>
<td>Specifies a delete operation.</td>
</tr>
<tr>
<td>-b, --bucket</td>
<td>Specifies the bucket on which to perform the operation.</td>
</tr>
<tr>
<td>--disable-crc64</td>
<td>Disables CRC-64. By default, ossutil enables CRC-64 during data transmission.</td>
</tr>
<tr>
<td>--upload</td>
<td>Uploads an object to OSS. This option is used with the <code>probe</code> command.</td>
</tr>
<tr>
<td>--part-size</td>
<td>Specifies the part size in bytes. By default, ossutil calculates the appropriate part size based on the object size. You can set this option to any positive integer if you need to optimize performance or are operating under special constraints.</td>
</tr>
<tr>
<td>--timeout</td>
<td>Specifies the timeout period of a signed URL request. Unit: seconds. Valid values: non-negative integers. Default value: 60.</td>
</tr>
<tr>
<td>-k, --access-key-secret</td>
<td>Specifies the AccessKey secret used to access OSS. This option value will overwrite the corresponding configurations in the configuration file.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--checkpoint-dir</td>
<td>Specifies the checkpoint directory path. Default value: .ossutil_checkpoint. 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, make sure that you have permissions to delete the specified directory.</td>
</tr>
<tr>
<td>--url</td>
<td>Specifies the URL of an object. This option is used with the probe command.</td>
</tr>
<tr>
<td>--marker</td>
<td>Specifies the bucket name, object name, or multipart upload ID from which you want to start listing.</td>
</tr>
<tr>
<td>-f, --force</td>
<td>Forces an operation without prompting the user for confirmation.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--snapshot-path</td>
<td>If you specify the --snapshot-path 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.</td>
</tr>
</tbody>
</table>

**Notice:**

- The --snapshot-path 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 the 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.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--loglevel</td>
<td>Specifies the log level. The default value is null, indicating that no log files are generated. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• info: generates prompt logs.</td>
</tr>
<tr>
<td></td>
<td>• debug: generates detailed logs that contain corresponding HTTP request and response information.</td>
</tr>
<tr>
<td>--storage-class</td>
<td>Specifies the storage method of an object. Valid values: Standard, IA, and Archive. Default value: Standard.</td>
</tr>
<tr>
<td>-i, --access-key-id</td>
<td>Specifies the AccessKey ID used to access OSS. This option value will overwrite the corresponding configurations in the configuration file.</td>
</tr>
<tr>
<td>-t, --sts-token</td>
<td>Optional. This option specifies the STS token used to access OSS. This option value will overwrite the corresponding configurations in the configuration file.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>--parallel</td>
<td>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.</td>
</tr>
<tr>
<td>--partition-download</td>
<td>Specifies the partition to download. The value of this option is in &quot;partition number: total number of partitions&quot; format. A value of 1:5 indicates that ossutil downloads partition 1 out of the total five partitions. Partitions are numbered starting from 1. 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.</td>
</tr>
<tr>
<td>--bucketname</td>
<td>Specifies the name of a bucket. This option is used with the probe command.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--local-host</td>
<td>Enter the local IP address of ossutil. If your computer has multiple IP addresses, you can specify this option so that ossutil can access OSS through the specified IP address. This option is used with the cp command.</td>
</tr>
<tr>
<td>--enable-symlink-dir</td>
<td>Specifies that the subdirectory to which the symbolic link points is uploaded. By default, subdirectories are not uploaded. The probe command can be used to check whether an object or a directory to which the symbolic link points is also a symbolic link.</td>
</tr>
<tr>
<td>--only-current-dir</td>
<td>Specifies that only objects in the current directory are uploaded, downloaded, or copied. The subdirectories in the current directory are ignored.</td>
</tr>
<tr>
<td>--disable-dir-object</td>
<td>Specifies that no OSS object is generated for the directory during object uploads.</td>
</tr>
<tr>
<td>--probe-item</td>
<td>Specifies the items to be checked by using the probe command. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• upload-speed: checks the upload bandwidth.</td>
</tr>
<tr>
<td></td>
<td>• download-speed: checks the download bandwidth.</td>
</tr>
<tr>
<td></td>
<td>• cycle-symlink: checks whether an object or a directory to which the symbolic link points in the local directory is also a symbolic link.</td>
</tr>
<tr>
<td>--redundancy-type</td>
<td>Specifies the data redundancy type of a bucket. Valid values: LRS (local redundant storage) and ZRS (zone-redundant storage). Default value: LRS.</td>
</tr>
<tr>
<td>--disable-encode-slash</td>
<td>Specifies that forward slashes (/) in the URL path are not encoded.</td>
</tr>
<tr>
<td>--disable-all-symlink</td>
<td>All objects in the subdirectory to which the symbolic link points and the subdirectory to which the symbolic link points are ignored during object uploads.</td>
</tr>
</tbody>
</table>

### 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 hotlink protection 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:

```bash
./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:

  ```bash
  ./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`:

```bash
./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 \(\text{rm}\) command, run the ossutil help \(\text{rm}\) 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 \(\text{ls}\) command, run the ossutil help \(\text{ls}\) command.

Example:

<table>
<thead>
<tr>
<th>CreationTime</th>
<th>Region</th>
<th>StorageClass</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-10-21 16:18:37 +0800 CST</td>
<td>oss-cn-hangzhou</td>
<td>Archive</td>
</tr>
<tr>
<td>oss://go-sdk-test-bucket-xyz-for-object</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
List your buckets by page

```bash
./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:

```bash
./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

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

Example:

```bash
./ossutil ls oss://ossutil-test
LastModifiedTime   Size(B) StorageClass  ETAG
2016-12-0115:06:37 +0800 CST  10363812 Standard
oss://ossutil-test/1
```

```bash
61DE142E5AFF9A6748707D4A77BFBCFB
```

```bash
2016-12-0115:06:42 +0800 CST  10363812 Standard
oss://ossutil-test/2
```

```bash
61DE142E5AFF9A6748707D4A77BFBCFB
```
**Object Storage Service**

**Tools / 3 ossutil**

2016-12-01 15:06:45 +0800 CST 10363812 Standard
61DE142E5AFF9A6748707D4A77BFBCFB 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
Object Name
2015-06-05 14:06:29 +0000 CST         201933      Standard 7E2F4A7F1AC9D2F0996E8332D5E5A5B41
oss://bucket1/dir1/obj1
2015-06-05 14:36:21 +0000 CST         201933      Standard 6185CA2E8EBB106A1B3A845EAE4174
oss://bucket1/obj1
2016-04-08 08:14:50:47 +0000 CST      6476984     Standard 4F16F6A7AC404CEC6B727FCC67779D6
oss://bucket1/sample.txt
Object Number is:3
```

```
InitiatedTime                     UploadID
ObjectName
2017-01-13 03:45:26 +0000 CST     15754AF7980C4DFB8193F190837520BB
oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST     2A1F9B4A95E341BD9285CC42BB950EE0
oss://bucket1/obj1
2017-01-13 03:45:25 +0000 CST     3998971ACAF94AD9AC48EAC1988BE863
oss://bucket1/obj2
2017-01-20 11:16:21 +0800 CST     A20157A7B2FEC4670626DAE0F4C0073C
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.
• **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
InitiatedTime                  UploadID                  ObjectName
2017-01-13 03:45:26 +0000 CST  15754AF7980C4DFB8193F190837520BB  oss://bucket1/obj1
2017-01-13 03:45:25 +0000 CST  3998971ACAF94AD9AC48EAC1988BE863  oss://bucket1/obj2
```
List the parts to be uploaded for all your objects

```bash
./ossutil getallpartsize oss://bucket
```

List the parts to be uploaded for a specified object

```bash
./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 #unique_64.

**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:

```bash
./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:

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

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

```bash
./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 #unique_62.
You can run the following command to add the encryption settings for a bucket:

```
./ossutil bucket-encryption --method put oss://bucket --sse-algorithm algorithmName [--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
<?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 hotlink protection rules

You can set the method parameter in the referer command to put, get, or delete to add, change, query, or delete the hotlink protection rule of a bucket. For more information, see #unique_77.

Note:

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

• Add or change the hotlink protection rule of a bucket

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

If no hotlink protection rule is set for the bucket, ossutil adds the specified hotlink protection rule. If a hotlink protection rule already exists, ossutil changes this hotlink protection rule to the specified hotlink protection 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 hotlink protection rule for a
bucket while disallowing the Referer field to be left unspecified:

```bash
test2.com --disable-empty-referer
```

- Obtain the hotlink protection rule of a bucket

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

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

- Delete the hotlink protection rule of a bucket

```bash
./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=TMP.xxxxxxxx5r9f1FV12y8_Qis6LUVmvoSCUSs7aboCCHtydQ9axN32Sn-UvyY3AAAwLAIUrYNLcQ87AKMEcE5O3AxxxxxxoCFAquRd Zy/yVFyq0W8QkGAN-bamUiQ&Signature=bIa4llbMbdrl7rwckr%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

```bash
./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

```bash
./ossutil probe --upload [file_name] --bucketname bucket-name [--object=obj ect_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 (×) appears following a step, then this step failed. If a multiplication sign (×) 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
0.699795(s) elapsed
```

- Upload a folder:

```
$./ossutil cp -r dir oss://ossutil-test
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:

  ```
  ```

  ■ Run the following command to modify the storage class of all objects in an existing folder to Standard:

  ```
  ```

**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 to change the storage class from Archive.
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 recursively 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/MSQK0xXSiyKBZog==
Content-Type : application/octet-stream
Etag : FB9BDB402FCC4902B4C574A2C8A059A2
Last-Modified : 2017-01-13 15:14:22 +0800 CST
Owner : aliyun
X-Oss-Hash-Crc64ecma : 12488808046134286088
X-Oss-Object-Type : Normal
```
· 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                    : D7257B62AA6A26D66686391037B7D61A
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
InitiatedTime                     UploadID
ObjectName
2017-01-13 03:45:26 +0000 CST     15754AF7980C4DFB8193F190837520BB
oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST     2A1F9B4A95E341BD9285CC42BB950EE0
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
LastModifiedTime                    Size(B)  StorageClass   ETAG
ObjectName
2015-06-05 14:36:21 +0000 CST        241561      Standard
6185CA2E8EB851061B3A845EAE4174        oss://bucket1/obj1/test.txt
2016-04-08 14:50:47 +0000 CST       6476984      Standard
4F16FDAE7AC404CE8B727FCC67779D6        oss://bucket1/obj1/sample.txt
Object Number is: 2
InitiatedTime                     UploadID
ObjectName
2017-01-13 03:45:26 +0000 CST     15754AF7980C4DFB8193F190837520BB
oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST     2A1F9B4A95E341BD9285CC42BB950EE0
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
LastModifiedTime                    Size(B)  StorageClass   ETAG
ObjectName
2015-06-05 14:06:29 +0000 CST        201933      Standard
7E2F4A7F1AC9D2F0996E8332D5EA5B41        oss://bucket1/dir1/obj1
```

Issue: 20200305
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.
```

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 recursively multipart uploadIds of oss:bucket1/ob(y or N)? y
Succeed: Total 4 uploadIds. Removed 4 uploadIds.
```

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 Architectures and configurations

ossimport is a tool used to migrate data to OSS. You can deploy ossimport on the local server or ECS instance in the cloud to migrate data stored locally or in other cloud storage systems to OSS.

ossimport has the following features:

- Supports a wide range of data sources including local data sources, Amazon Simple Storage Service (Amazon S3), Azure Blob, Qiniu Cloud Object Storage (KODO), Baidu Object Storage (BOS), UPYUN Storage Service (USS), Tencent Cloud Object Service (COS), Kingsoft Standard Storage Service (KS3), HTTP, and OSS. These sources can be expanded as needed.
- Supports standalone and distributed modes. The standalone mode is easy to deploy and use. The distributed mode is suitable for large-scale data migration.
- Supports resumable data transfer.
- Supports throttling.
- Supports migration of objects after a specified point in time or of objects whose names contain a specified prefix.
- Supports the upload and download of data in parallel.

Runtime environment

ossimport can be deployed to Linux or Windows that meets the following requirements:

- Windows 7 or later versions
- Latest Linux version
- Java 1.7 or later versions

Notice:
ossimport cannot be deployed to distributed mode if deployed to Windows.

Deployment modes

ossimport supports the standalone and distributed modes.
- The standalone mode is sufficient for small-scale migration of data smaller than 30 TB. You can deploy ossimport on any machine locally or in the cloud. Download

- The distributed mode is suitable for migration of data larger than 30 TB. You can deploy ossimport on multiple machines locally or in the cloud as needed. Download

Note:
To save time for migration of huge amounts of data, deploy ossimport on an ECS instance in the same region as your OSS bucket. Use a leased line to attach the server that stores source data to Alibaba Cloud VPC. Migration efficiency is greatly improved when the internal network is used to migrate data from ECS instances to OSS.

Standalone mode

Master, Worker, Tracker, and Console are packaged as ossimport2.jar. They run on a machine. The system has only one Worker.

The file structure in standalone mode is as follows:

```
ossimport
  └── bin
      └── ossimport2.jar  # The JAR package that contains the Master, Worker, Tracker, and Console modules.
  └── conf
      └── local_job.cfg   # The Job configuration file for the standalone mode.
      └── sys.properties  # The configuration file that contains system parameters.
  ├── console.bat         # The command-line tool in Windows used to run tasks step by step.
  ├── console.sh          # The command-line tool in Linux used to run tasks step by step.
  ├── import.bat          # The script that automatically imports files based on the conf/local_job.cfg configuration file in Windows. The configuration file contains parameters that specify operations involved in data migration, such as start, migration, verification, and retry.
  └── import.sh           # The script that automatically imports files based on the conf/local_job.cfg configuration file in Linux. The configuration file contains parameters that specify operations involved in data migration, such as start, migration, verification, and retry.
  └── logs                # The directory that contains logs.
```
import.bat and import.sh are scripts that automatically import files based on the configuration file. You can run these tools after you modify the local_job.cfg configuration file.

console.bat and console.sh are command-line tools used to run commands step by step.

Run scripts or commands in the ossimport directory. These scripts and the *.bat /*.sh file are at the same directory level.

Distributed mode

The ossimport architecture in distributed mode consists of Master and Worker. The structure is as follows:

<table>
<thead>
<tr>
<th>Master</th>
<th>Job</th>
<th>Console</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TaskTracker</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Task</td>
<td>Task</td>
</tr>
<tr>
<td>Worker</td>
<td>Worker</td>
<td>Worker</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Master    | Splits a job into multiple tasks by data size and number of files. The data size and number of files can be configured in the sys.properties file. The splitting process is as follows:  
1. Master traverses the full list of files to be migrated from the local source or cloud storage system.  
2. Master splits a job into multiple tasks by data size and number of files. Each task is responsible for the migration or verification of a portion of files. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Worker    | • Migrates files and verifies data for tasks. It pulls the specific file from the data source and uploads the file to the specified directory in OSS. You can specify the data source to be migrated and the OSS configuration in the job.cfg or local_job.cfg configuration file.  
• Supports throttling and specifies the number of concurrent tasks for data migration. You can configure the settings in the sys.properties configuration file. |
| TaskTracker | Distributes tasks and tracks task statuses. It is abbreviated to Tracker. |
| Console  | Interacts with users and receives command display results. It supports system management commands such as deploy, start, and stop, and job management commands such as submit, retry, and clean. |
| Job      | Serves as the data migration jobs submitted by users. One job corresponds to one configuration file job.cfg. |
| Task     | Migrates a portion of files. A job can be divided into multiple tasks by data size and number of files. The minimal unit for dividing a job into tasks is a file. One file cannot be used to split into multiple tasks. |

In distributed mode, multiple Workers can be started to migrate data. Tasks are evenly allocated to Workers. One Worker can run multiple tasks. Only one Worker can be started on each machine. Master and Tracker are started on the machine where the first Worker specified by workers resides. Console must also run on this machine.

The file structure in distributed mode is as follows:

```
ossimport
  └── bin
      ├── console.jar  # The JAR package for the Console module.
      ├── master.jar   # The JAR package for the Master module.
      └── tracker.jar  # The JAR package for the Tracker module.
  └── conf
      ├── job.cfg     # The Job configuration file template.
      └── sys.properties # The configuration file that contains system parameters.
  └── workers       # The list of Workers.
  └── console.sh    # The command-line tool. Currently, only Linux is supported.
```
Configuration files

Two configuration files `sys.properties` and `local_job.cfg` are provided in standalone mode. Three configuration files `sys.properties`, `job.cfg`, and `workers` are provided in distributed mode. Configurations items in the `local_job.cfg` and `job.cfg` configuration files are the same except for their file names. The `workers` configuration file is exclusive to the distributed mode.

- **sys.properties**: the system parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>workingDir</td>
<td>The working directory.</td>
<td>The directory to which the tool package is decompressed. Do not modify this parameter in standalone mode. Working directories of each machine in distributed mode must be the same.</td>
</tr>
</tbody>
</table>
| workerUser    | The SSH username used to log on to the machine where Worker resides. | - If privateKeyFile is configured, privateKeyFile is prioritized.  
- If privateKeyFile is not configured, use workerUser and workerPassword.  
- Do not modify this parameter in standalone mode. |
<p>| workerPassword| The password of the SSH username used to log on to the machine where Worker resides. | Do not modify this parameter in standalone mode. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>privateKeyFile</td>
<td>The path of the private key file.</td>
<td>- If you have established an SSH channel, you can specify this parameter. Otherwise, leave this parameter unspecified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If privateKeyFile is configured, privateKeyFile is prioritized.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If privateKeyFile is not configured, use workerUser and workerPass word.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Do not modify this parameter in standalone mode.</td>
</tr>
<tr>
<td>sshPort</td>
<td>The SSH port.</td>
<td>The default value is 22. We recommend that you retain the default value. Do not modify this parameter in standalone mode.</td>
</tr>
<tr>
<td>workerTaskThreadNum</td>
<td>The maximum number of threads for Worker to run tasks.</td>
<td>- This parameter is related to the machine memory and network. We recommend that you set this parameter to 60.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The value can be increased. For example, you can set this parameter to a greater value such as 150 for physical machines. If the network bandwidth is already full, do not further increase the value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If the network is poor, lower the value as needed, such as 30. This way, you can avoid the timeout of a large number of requests and request competition for bandwidth.</td>
</tr>
<tr>
<td>workerMaxThroughput(KB/s)</td>
<td>The throttling of data migration for Worker.</td>
<td>This value can be used for throttling. The default value is 0, indicating that no throttling is imposed.</td>
</tr>
<tr>
<td>dispatcherThreadNum</td>
<td>The number of threads for task distribution and status confirmation of Tracker.</td>
<td>If there are no special requirements, retain the default value.</td>
</tr>
</tbody>
</table>
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>workerAbortWhenUncatchedException</td>
<td>Indicates whether to skip or terminate a task if an unknown error occurs.</td>
<td>By default, unknown errors are skipped.</td>
</tr>
<tr>
<td>workerRecordMd5</td>
<td>Indicates whether to use metadata <code>x-oss-meta-md5</code> to record the MD5 values of files to be migrated. By default, MD5 values are not recorded.</td>
<td>This parameter value is used to verify data integrity.</td>
</tr>
</tbody>
</table>

- **job.cfg**: The configurations for data migration tasks. Configuration items in the `local_job.cfg` and `job.cfg` configuration files are the same except for the file names.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobName</td>
<td>The name of the job. The value is of the String type.</td>
<td>- The unique identifier of the job. The naming convention is as follows: The name can contain letters, digits, underscores (_), and hyphens (-). The name must be 4 to 128 characters in length. You can submit multiple jobs of different names. - If you submit a job with the same name as an existing 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 existing job with the same name.</td>
</tr>
<tr>
<td>jobType</td>
<td>The type of the job. The value is of the String type.</td>
<td>Valid values: import and audit. Default value: import. - import: runs the data migration job and verifies the migration data for consistency. - audit: only verifies data consistency.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Purpose</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| isIncremental      | Indicates whether to enable the incremental migration mode. The value is of the Boolean type. | - Default value: false  
- If this parameter is set to true, incremental data is rescanned at the interval specified by incrementalModeInterval in seconds and synchronized to the OSS. |
| incrementalModeInterval | The synchronization interval in seconds in the incremental mode. The value is of the Integer type. | This parameter is valid when isIncremental is set to true. The minimum configurable interval is 900s. We recommend that you do not set it to a value smaller than 3600s. If you set this parameter to a smaller value, a large number of requests are wasted, resulting in extra system overheads. |
| importSince        | The time in seconds after which to migrate data. The value is of the Integer type. | - The timestamp follows the UNIX time format. It is the number of seconds that have elapsed since 00:00:00 Thursday, 1 January 1970. You can run the date +%s command to obtain the seconds.  
- The default value is 0, indicating that all data is migrated. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>srcType</td>
<td>The source type for synchron</td>
<td>The following source types are supported:</td>
</tr>
<tr>
<td></td>
<td>ization. The value is of t</td>
<td>- local: migrates data from a local file to OSS. To specify this option,</td>
</tr>
<tr>
<td></td>
<td>he String type. Note that t</td>
<td>specify srcPrefix and ignore srcAccessKey, srcSecretKey, srcDomain, and s</td>
</tr>
<tr>
<td></td>
<td>he value is case-sensitive.</td>
<td>rcBucket.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- oss: migrates data from one OSS bucket to another bucket.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- qiniu: migrates data from KODO to OSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- bos: migrates data from BOS to OSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ks3: migrates data from KS3 to OSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- s3: migrates data from Amazon S3 to OSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- youpai: migrates data from USS to OSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- http: migrates data from HTTP sources to OSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- cos: migrates data from COS to OSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- azure: migrates data from Azure Blob to OSS.</td>
</tr>
<tr>
<td>srcAccessKey</td>
<td>The AccessKey ID used to a</td>
<td>- If srcType is set to oss, qiniu, baidu, ks3, or s3, specify the Access</td>
</tr>
<tr>
<td></td>
<td>ccess the source. The val</td>
<td>Key ID used to access the source.</td>
</tr>
<tr>
<td></td>
<td>ue is of the String type.</td>
<td>- If srcType is set to local or http, ignore this parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If srcType is set to youpai or azure, specify the username used to ac</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
</table>
| srcSecretKey  | The AccessKey secret used to access the source. The value is of the String type. | - If srcType is set to oss, qiniu, baidu, ks3, or s3, specify the AccessKey secret used to access the source.  
- If srcType is set to local or http, ignore this parameter.  
- If srcType is set to youpai, specify the operator password used to access the source.  
- If srcType is set to azure, specify the account key used to access the source. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
</table>
| srcDomain | The source endpoint. | - If srcType is set to local or http, ignore this parameter.  
- If srcType is set to oss, enter the endpoint obtained from the OSS console. The endpoint is a second-level domain without the bucket name as the prefix.  
- If srcType is set to qiniu, enter the domain name corresponding to the bucket obtained from the KODO console.  
- If srcType is set to bos, enter the BOS domain name. Example: http://bj.bcebos.com or http://gz.bcebos.com.  
- If srcType is set to S3, enter the domain name to access the region where Amazon S3 resides.  
- If srcType is set to youpai, enter the USS domain name such as automatic identification of the optimal path of http://v0.api.upyun.com, telecommunication line http://v1.api.upyun.com, China Unicom or China Netcom line http://v2.api.upyun.com, or China Mobile or China Railcom line http://v3.api.upyun.com.  
- If srcType is set to cos, enter the domain name to access the COS bucket located in the region. Example: Guangzhou (South China), Tianjin (North China), or Shanghai (East China).  
- If srcType is set to azure, enter the endpoint suffix in the Azure Blob connection string. Example: core.chinacloudapi.cn. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
</table>
| srcBucket   | The name of the source bucket or container.       | - If srcType is set to local or http, ignore this parameter.  
- If srcType is set to azure, enter the name of the container.  
- In other cases, enter the name of the source bucket.                                                                 |                                                                                                                                                                                                                                                                                                                                                     |
| srcPrefix   | The source prefix.                                | - If srcType is set to local, enter the full path that is enclosed in a pair of forward slashes (/). If the path contains two or more directory levels, separate each directory with one forward slash (/). Example: c:/example/ or /data/example/.  
- If srcType is set to oss, qiniu, bos, ks3, youpai, or s3, enter the prefix for objects to be synchronized. The prefix excludes bucket names. Example of the prefix: data/to/oss/.  
- To synchronize all files, leave srcPrefix unspecified.                                                                                                                                                                                                                                                                                        |
|             |                                                   | **Notice:**  
Paths such as c:/example//, /data//example/, and /data/example// are invalid.                                                                                                                                                                                                                                                                                                                                                                           |
| destAccessKey | The AccessKey ID used to access the destination.   | The value is of the String type.  
To obtain the AccessKey ID to access the OSS bucket, log on to Alibaba Cloud Management Console.                                                                                                                                                                                                                                                                                                                                                                      |
| destSecretKey | The AccessKey secret used to access the destination. | The value is of the String type.  
To obtain the AccessKey secret used to access the OSS bucket, log on to Alibaba Cloud Management Console.                                                                                                                                                                                                                                                                                                                                                                       |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destDomain</td>
<td>The destination endpoint.</td>
<td>To obtain the second-level domain without the bucket name as the prefix, log on to Alibaba Cloud Management Console.</td>
</tr>
<tr>
<td></td>
<td>The value is of the String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>type.</td>
<td></td>
</tr>
<tr>
<td>destBucket</td>
<td>The destination bucket.</td>
<td>The name of the OSS bucket. The name cannot end with a forward slash (/).</td>
</tr>
<tr>
<td></td>
<td>The value is of the String</td>
<td></td>
</tr>
<tr>
<td></td>
<td>type.</td>
<td></td>
</tr>
<tr>
<td>destPrefix</td>
<td>The destination prefix.</td>
<td>- The destination prefix. If you retain the default value, the migrated objects are stored in the destination bucket.</td>
</tr>
<tr>
<td></td>
<td>The value is of the String</td>
<td>- To synchronize data to a specified directory in OSS, end the prefix with a forward slash (/). Example of the prefix: data/in/oss/.</td>
</tr>
<tr>
<td></td>
<td>type.</td>
<td>- Note that OSS object names cannot start with a forward slash (/). Do not start the destination prefix with a forward slash (/).</td>
</tr>
<tr>
<td></td>
<td>The default value is null.</td>
<td>- A local file stored in the path with the format of srcPrefix+relativePath is migrated to destDomain/destBucket/destPrefix+relativePath in OSS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- An object stored in the cloud in the path with the format of srcDomain/srcBucket/srcPrefix+relativePath is migrated to destDomain/destBucket/destPrefix+relativePath in OSS.</td>
</tr>
<tr>
<td>taskObject</td>
<td>The maximum number of files</td>
<td>This configuration option affects the concurrency of jobs to run. In most cases, this parameter is set based on the formula: Value = Total number of files/Total number of Workers/Number of migration threads (workerTaskThreadNum). The maximum value is 50000. If the total number of files is unknown, use the default value.</td>
</tr>
<tr>
<td>CountLimit</td>
<td>in each Task. The value is</td>
<td></td>
</tr>
<tr>
<td></td>
<td>of the Integer type. The</td>
<td></td>
</tr>
<tr>
<td></td>
<td>default value is 10000.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Purpose</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>taskObject SizeLimit</td>
<td>The maximum data size in bytes for each task. The value is of the Integer type. The default value is 1 GB.</td>
<td>This configuration option affects the concurrency of jobs to run. In most cases, this parameter is set based on the formula: Value = Total data size /Total number of Workers/Number of migration threads (workerTask ThreadNum). If the total data size is unknown, use the default value.</td>
</tr>
<tr>
<td>isSkipExistFile</td>
<td>Indicates whether to skip the existing objects during data migration. The value is of the Boolean type.</td>
<td>If this parameter is set to true, the objects are skipped based on the size and LastModifiedTime. If this parameter is set to false, the existing objects are overwritten. The default value is false. This option is invalid when jobType is set to audit.</td>
</tr>
<tr>
<td>scanThreadCount</td>
<td>The number of threads that scan files in parallel. The value is of the Integer type.</td>
<td>This configuration option is related to file scanning efficiency. If there are no special requirements, retain the default value.</td>
</tr>
</tbody>
</table>
| maxMultiThreadScanDepth | The maximum allowable depth of directories for parallel scanning. The value is of the Integer type. | - Default value 1 indicates parallel scanning on top-level directories.  
- If there are no special requirements, retain the default value. A large value may cause task failures. |
<p>| appId                 | The application ID (account number ) of COS. The value is of the Integer type. | This parameter is valid when srcType is set to cos. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Description</th>
</tr>
</thead>
</table>
| httpListFilePath | The absolute path of the HTTP list file. The value is of the String type. | - This parameter is valid when srcType is set to http. When the source is accessed through an HTTP link, you must provide the absolute path of the file that contains the HTTP link. Example: c:/example/http.list.  
- The HTTP link in the file must be divided into two columns separated with spaces, indicating the prefix and the relative path in OSS after the upload. For example, the c:/example/http.list file can contain two rows: http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/aa/bb.jpg and http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/cc/dd.jpg. The object names in OSS after the objects are migrated are in destPrefix+bb.jpg and destPrefix+cc/dd.jpg formats. |

- workers: exclusive to the distributed mode. Each IP address is separated with a line break. Example:

  192.168.1.6  
  192.168.1.7  
  192.168.1.8

- In the preceding configuration, 192.168.1.6 is in the first row, which must be master. 192.168.1.6 is the IP address of the machine where Master, Worker, and TaskTracker are started. Console also runs on this machine.  
- Make sure that the username, logon mode, and working directory in multiple-Worker mode are the same.

Configuration file examples

The following table describes the configuration file of a data migration task in distributed mode. The name of the configuration file in standalone mode is local_job.cfg, which contains the same configuration items as those in distributed mode.
<table>
<thead>
<tr>
<th>Migration type</th>
<th>Configuration file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate data locally to OSS</td>
<td><code>job.cfg</code></td>
<td>srcPrefix specifies an absolute path that ends with a forward slash (<code>/</code>). Example: <code>D:/work/oss/data/</code> or <code>/home/user/work/oss/data/</code>.</td>
</tr>
<tr>
<td>Migrate data from KODO to OSS</td>
<td><code>job.cfg</code></td>
<td>You can leave srcPrefix and destPrefix unspecified. To specify these parameters, end the prefixes with a forward slash (<code>/</code>). Example: <code>destPrefix=docs/</code>.</td>
</tr>
<tr>
<td>Migrate data from BOS to OSS</td>
<td><code>job.cfg</code></td>
<td>You can leave srcPrefix and destPrefix unspecified. To specify these parameters, end the prefixes with a forward slash (<code>/</code>). Example: <code>destPrefix=docs/</code>.</td>
</tr>
<tr>
<td>Migrate data from AWS S3 to OSS</td>
<td><code>job.cfg</code></td>
<td>For more information, see AWS Service Endpoints.</td>
</tr>
<tr>
<td>Migrate data from USS to OSS</td>
<td><code>job.cfg</code></td>
<td>Set srcAccessKey and srcSecretKey to the operator account and password.</td>
</tr>
<tr>
<td>Migrate data from COS to OSS</td>
<td><code>job.cfg</code></td>
<td>Set srcDomain based on V4. Example: <code>srcDomain=sh</code>. You can leave srcPrefix unspecified. To specify this parameter, start and end the prefix with a forward slash (<code>/</code>). Example: <code>srcPrefix=/docs/</code>.</td>
</tr>
<tr>
<td>Migrate data from Azure Blob to OSS</td>
<td><code>job.cfg</code></td>
<td>Set srcAccessKey and srcSecretKey to the storage account and access key. Set srcDomain to the endpoint suffix in the Azure Blob connection string. Example: <code>core.chinacloudapi.cn</code>.</td>
</tr>
<tr>
<td>Migrate data between buckets in OSS</td>
<td><code>job.cfg</code></td>
<td>This method is suitable for data migration between different regions, different storage classes, and objects whose names have different prefixes. We recommend that you deploy your service on ECS and use the domain name for access over the internal network to minimize the traffic cost.</td>
</tr>
</tbody>
</table>
Advanced settings

- **Time-specific throttling**

  In the `sys.properties` configuration file, `workerMaxThroughput(KB/s)` specifies the upper throttling limit for Worker. To configure throttling for business such as throttling for the source and network limits, set this parameter to a value smaller than the maximum available bandwidth for the machine based on business requirements. After the modification is complete, restart the service for the modification to take effect.

  In distributed mode, modify the `sys.properties` configuration file in the `$OSS_IMPORT_WORK_DIR/conf` directory for each Worker. Restart the service.

  To implement time-specific throttling, modify the `sys.properties` configuration file as scheduled through crontab and restart the service for the modification to take effect.

- **Modify the number of concurrent tasks**

  - In the `sys.properties` configuration file, `workerTaskThreadNum` specifies the number of concurrent tasks run by Worker. If the network is poor and Worker has to process a large number of tasks, timeout errors are returned. To resolve this issue, modify the configuration by reducing the number of concurrent tasks and restart the service.

  - In the `sys.properties` configuration file, `workerMaxThroughput(KB/s)` specifies the upper throttling limit for Worker. To configure throttling for business such as throttling for the source and network limits, set this parameter to a value smaller than the maximum available bandwidth for the machine based on business requirements.

  - In the `job.cfg` configuration file, `taskObjectCountLimit` specifies the maximum number of files in each task. The default value is 10000. This parameter configuration affects the number of tasks. The efficiency of implementing concurrent tasks may degrade if you set this parameter to a small value.

  - In the `job.cfg` configuration file, `taskObjectSizeLimit` specifies the maximum data size for each task. The default value is 1 GB. This parameter
configuration affects the number of tasks. The efficiency of implementing concurrent tasks may degrade if you set this parameter to a small value.

Note:

- We recommend that you specify the configuration file parameters before you start the migration.
- After you modify parameters in the sys.properties configuration file, restart the server for migration to make the modification take effect.
- After job.cfg is submitted, parameters in the job.cfg configuration file cannot be modified.

- Data verification without migration

To specify that ossimport only verifies data without migrating data, set the job.cfg or local_job.cfg configuration file. Set jobType to audit instead of import. Configurations of other parameters are the same as those for data migration.

- Incremental data migration mode

The incremental data migration mode allows you to migrate existing data after the migration task is started and migrate incremental data at intervals. The first data migration task to migrate existing data is started after you submit the task. Incremental data is migrated at intervals. The incremental data migration mode is suitable for data backup and synchronization.

The following configuration items are available for the incremental data migration mode:

- In the job.cfg configuration file, isIncremental specifies whether to enable the incremental data migration mode. true indicates that the incremental data migration mode is enabled. false indicates that the incremental data migration mode is disabled. The default value is false.

- In the job.cfg configuration file, incrementalModeInterval indicates the interval in seconds at which incremental data migration is implemented. This configuration takes effect if you specify isIncremental=true. The minimum configurable value is 900s. We recommend that you do not set this parameter to a value smaller than 3600. If you set this parameter to a smaller value, a large number of requests are wasted, resulting in extra system overheads.
Object Storage Service

Tools / 4 ossimport

- Filtering conditions for source objects
  You can set filtering conditions to migrate objects that meet specified conditions. ossimport allows you to specify prefixes and last modification time.

  - In the job.cfg configuration file, srcPrefix specifies the prefix of the source objects. The default value is null.

    ■ If you specify srcType=local, enter the local directory path. Enter the full path that is enclosed in a pair of forward slashes (/). If the path contains two or more directory levels, separate each directory with one forward slash (/). Example: c:/example/ or /data/example/.

    ■ If you set srcType to oss, qiniu, bos, ks3, youpai, or s3, enter the name prefix of source objects. Example: data/to/oss/. To migrate all objects, leave SrcPrefix unspecified.

  - In the job.cfg configuration file, importSince specifies the last modification time in seconds for source objects, importSince specifies the timestamp that follows the UNIX time format. It is the number of seconds that have elapsed since 00:00:00 Thursday, 1 January 1970. You can run the date +%s command to obtain the seconds. The default value is 0, indicating that all data is migrated. The incremental data migration mode is valid only for the first full migration. The non-incremental mode is valid for the entire migration task.

    ■ If the LastModified Time values of objects are earlier than the importSince value, objects are not migrated.

    ■ If the LastModified Time values of objects are later than the importSince value, objects are migrated.

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
Object Storage Service

Tools 4 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 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 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
└── README.md  # Description documentation. We recommend that you carefully read the documentation 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, workerUserName, workerPassword, and privateKeyFile.

- **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_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 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
```
Object Storage Service

Tools / 4 ossimport

- 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
    - local_test # Store the job running status
      - checkpoints # The checkpoint record of the master splitting the job to tasks
        - 0 034DC9DD2860B0C64F92E7.cpt
        - dispatched # Tasks that have been assigned to the workers but haven't been fully run
        - localhost
        - failed_tasks # Tasks that failed to run
        - pending_tasks # Tasks that have not been assigned
        - succeed_tasks # Tasks that run successfully
          - A41506C07BF1DF2AE4CB7E31756B93F_1499744514501@
  - localhost
  - view the error causes in the log
    - audit.log # The task running log. You can view the error list
    - DONE # Mark of successful tasks
    - error.list # The task error list. You can view the error file list
    - p to view the error file list
    - STATUS # The task status marker file. The content is Failed or Completed
    - TASK # The task description
  - 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/${JobName}/failed_tasks/${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 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 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 configuration file
│   ├── sys.properties # Configuration file of the system running parameters
│   └── workers      # Worker list
│            └── console.sh # The command line tool. Currently it only supports Linux
├── logs        # Log directory
└── README.md   # Description documentation. 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:
object storage service

Tools / 4 ossimport

| └── pending_tasks    # Tasks that have not been assigned
|     └── succeed_tasks # Tasks that run successfully
|         └── A41506C07BF1DF2A3EDB4CE31756B93F_1499668462358@
|             └── 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 description
|                     └── worker  # state of the task being run by the worker. After running, tasks are managed by the master
|                         └── jobs
|                             └── 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 Troubleshooting

This topic describes the causes and solutions for common problems you may encounter when you use ossimport.

**Note:**
All ossimport commands mentioned in this topic are shortened and the complete form must be used in practice.

- Add `console.bat` to Windows commands. For example, change `submit` to `console.bat submit`.
- Add `bash console.sh` to Linux commands. For example, change `submit` to `bash console.sh submit`.

Common problems about migration failures

If a migration job fails, we recommend that you view the migration failure log to identify the cause. After resolving the problems, you can run the `retry` command to migrate the files again. The path of the migration failure log is `master/jobs/${JobName}/failed_tasks/${TaskName}/audit.log`.

- The job state is displayed as failed when you run the `stat` command.

**Solution:** Run the `stat` command to view the job state. If `JobState` is failed, the migration job fails. After the migration job is complete, run the `retry` command to migrate the files again.

- Some files fail to be migrated and retrying the migration also fails.

**Solution:**

1. View the list of failed files from `master/jobs/${JobName}/failed_tasks/${TaskName}/error.list` to obtain their relative paths.
2. Check whether you are authorized to access these files, whether these files are deleted, whether they are symbolic links, and whether the file names are garbled.
3. After the preceding problems are resolved, run the `retry` command to migrate the files again.
- "The bucket you are attempting to access must be addressed using the specified endpoint" is displayed in the migration failure log.

Exception: com.aliyun.oss.OSSException: The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.

Analysis: The value of srcDomain or destDomain is invalid. Enter valid endpoints based on the endpoints listed in Regions and endpoints.

- "The request signature we calculated does not match the signature you provided" is displayed in the migration failure log.

Exception: com.aliyun.oss.OSSException: The request signature we calculated does not match the signature you provided. Check your key and signing method.

Analysis: The values of destAccessKey and destSecretKey are invalid. Enter a valid AccessKey pair.

- "The bucket name "xxx/xx" is invalid" is displayed in the migration failure log.

java.lang.IllegalArgumentException: The bucket name "xxx/xx" is invalid. A bucket name 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.

Analysis: Check whether the value of destBucket is valid. The bucket name cannot contain forward slashes (/) and paths.

- Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 timed out" is displayed in the migration failure log.

Unable to execute HTTP request: Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 timed out

Analysis: The connection timeout error is returned because the configuration file uses the internal endpoint of OSS, but the device that was used to migrate data
is not an ECS instance or is not an ECS instance that is in the same region as OSS instances. The internal endpoint supports access only from ECS instances that are located in the same region as OSS instances.

Solution:

- Set the domain name to a public endpoint in the configuration file. Delete the migration job and submit the job again.
- Use an ECS instance that is located in the same region as OSS instances to implement the migration job.

- "The specified bucket is not valid" is displayed in the migration failure log.

```
com.aliyun.oss.OSSEnException: The specified bucket is not valid.
[ErrorCode]: InvalidBucketName
[RequestId]: 57906B4DD0EBAB0FF553D661
[HostId]: you-bucket.you-bucketoss-cn-hangzhou-internal.aliyuncs.com
```

Analysis: The value of destDomain must be the endpoint of the region where the bucket is located, rather than the second-level domain name that contains the bucket name. For example, if the bucket is located in China (Beijing), enter oss-cn-beijing.aliyuncs.com. For more information, see Configuration file examples.

- "Unable to execute HTTP request: The Difference between ... is too large" is displayed in the migration failure log.

```
Unable to execute HTTP request: The Difference between the request time and the current time is too large.
[ErrorCode]: RequestTimeTooSkewed
```
[RequestId]: xxxxxxx

Analysis: This error may result from either of the following reasons.

- The difference between the system time of the machine or device sending the request and the system time adopted by OSS exceeds 15 minutes. This is the most common situation.
- A large number of requests are sent at the same time, resulting in a high CPU utilization and slow concurrent uploads.

Solution:

- Set the system time of the machine or device that is sending the request to the same time adopted by OSS.
- If the error is caused by high concurrency, you can reduce the number of requests that are sent at a time. You can set `workerTaskThreadNum` of the `sys.properties` file to a smaller value.

"No route to host" is displayed in the migration failure log.

Analysis: This error is returned when the network connection fails because of local firewalls or iptables.

Solution: Run the ping command to check whether the network connection between the source and destination instances is normal.

- If the network connection is normal, you can check if there are restrictions configured by the computer firewalls and local firewalls. You can also try disabling the firewall.
- If the network connection is abnormal, troubleshoot the problem and try again.
"Unknown http list file format" is displayed in the migration failure log when you migrate files over HTTP.

Analysis: This error is returned when the format of the specified HTTP list file or its content is invalid.

Solution:
- You can convert the format of files that are copied from a different operating system by using commands such as mac2unix and doc2unix in Linux and tools such as notepad in Windows.
- If the format of the HTTP list file content is invalid, change the content to the valid format. For more information about the format of the HTTP list file content, see the "Configuration files" section in Architectures and configurations.

"The object key "/xxxxx.jpg" is invalid" is displayed in the migration failure log.

Solution:
- Check whether srcPrefix is set to a folder but its value does not end with a forward slash (/). If you set the value to a folder, it must end with a forward slash (/).
- Check whether the value of destPrefix starts with a forward slash (/) or a backslash (/). The value of destPrefix cannot start with a forward slash (/) or a backslash (/).

Common problems when running a migration job

If a problem occurs during the migration, you can first view the running log. When ossimport is deployed in standalone mode, the running log file path is logs/ossimport2.log. When ossimport is deployed in distributed mode, the running log file path is logs/import.log.

"UnsupportedClassVersionError" is returned when a command is run.

Exception in thread "main" java.lang.UnsupportedClassVersionError: com/aliyun/ossimport2/OSSImport2 : Unsupported major.minor version 51.0
at java.lang.ClassLoader.defineClass1(Native Method)
at java.lang.ClassLoader.defineClassCond(ClassLoader.java:631)
at java.lang.ClassLoader.defineClass(ClassLoader.java:615)
Analysis: The Java version is earlier than the required version. Update it to version 1.7 or 1.8.

- "InvocationTargetException" is returned when you use the submit command to submit a job.

```java
Exception in thread "main" java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0(NativeMethod)
    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.NullPointerException
    at com.aliyun.ossimport2.config.JobConfig.load(JobConfig.java:44)
    at com.aliyun.ossimport2.OSSImport2.doSubmitJob(OSSImport2.java:289)
    at com.aliyun.ossimport2.OSSImport2.main(OSSImport2.java:120)
```

Analysis: This error is returned because the system identifies that some configuration items are deleted or commented out from the configuration file.

Solution: Restore the configuration items that are deleted or commented out from the configuration file. Leave configurations items unspecified after the equal sign (=) instead of deleting them. For configuration file examples, see Configuration file examples.

- "com.aliyun.oss.ClientException: Unknown" is displayed in the running log.

```java
com.aliyun.oss.ClientException: Unknown
    [ErrorCode]: NonRepeatableRequest
    [RequestId]: Cannot retry request with a non-repeatable request entity. The cause lists the reason the original request failed.
```

Analysis: "com.aliyun.oss.ClientException: Unknown" or "SocketTimeoutException" error is returned when the server is at full
bandwidth. In this case, ossimport automatically retries. If the job still fails, run the `retry` command to migrate the files again.

- "Too many open files" is displayed in the running log in Linux.

  Solution: Run the `ulimit -n` command to view the limit of handles in Linux.

  - If the handle limit is greater than 10,000, use the `ulimit -n 65536` command to increase the value and restart the process.
  - If handle values are greater than 100,000, use the `sudo losf -n` command to check which processes have enabled the handles. You can evaluate these processes and kill the unnecessary processes to release the handles.

- The migration job exits in seconds after starting in Windows

  Analysis:

  - Java is not installed or a Java version earlier than 1.7 is used.
  - Configuration file error.

  Solution:

  - Install the supported Java version.
  - Follow examples to edit configuration files. For more information about configuration file examples, see Configuration file examples.

- After a job is submitted by using the `submit` command, "no jobs is running or finished" is displayed when you use the `stat` command to view the job status.

```
bash console.sh stat
[WARN] List files dir not exist : /home/<user>/ossimport/workdir/master/jobs/
```
Analysis: A job starts running when the service is started and the job is submitted. In this case, you can use the `stat` command to view the job status.

- If you only submitted the job by using the `submit` command, but did not start the service by using the `start` command, "no jobs is running or finished" is displayed. You must use the `start` command to start the service.

- If the service is started and the job has just been submitted, Master must scan the file list first. Because the job has not been generated and distributed at this time, this error is displayed in the logs.

- If the error still occurs after the service has been started and the task has been submitted for a period of time, check whether the process exits unexpectedly after it starts. When ossimport is deployed in standalone mode, view the log file from `logs/ossimport2.log`. When ossimport is deployed in distributed mode, view the log file from `logs/ossimport.log`. Find the cause of the exception, resolve it, and then start the service process.

  "scanFinished: false " is displayed when you use the `stat` command to check the job status.

Solution: Check whether the total number of tasks is increasing.

- If the total number of tasks is increasing, the file list of the migration job is listing new files. It is normal that this error is returned in this case.

- If the incremental data migration mode is enabled, `scanFinished` is not true, and the number of tasks does not change, ossimport scans the file list at specified intervals to check for new or modified files.

- If the incremental data migration mode is disabled and the number of tasks does not increase, check the running log for exceptions. When ossimport is deployed in standalone mode, view the log file from `logs/ossimport2.log`. When ossimport is deployed in distributed mode, view the log file from `logs/ossimport.log`. Find the cause of the exception, resolve it, and then start the service process.

- The process in Linux does not respond, but no exception is output in logs

Analysis: If the system has less than 2 GB of available memory, the process may be killed due to insufficient memory. You can check whether the `dmesg` log
contains records about the processes that were killed because of insufficient memory.

- How do I restart a service after the process stops responding or is killed?

All submitted jobs have breakpoint records. If you do not clear the original job by using the `clean` command, you can directly use the `start` command to start the service without submitting the job again.

- How do I upload files whose names are garbled to OSS in Linux?

Solution:

1. Check the encoding format of the garbled names.
2. Use the `export LANG="<your file name encode>"` command to parse these encoded names.
3. Use the `clean` command to clear the original job, and then use the `submit` command to submit the job again.

- "java.nio.file.AccessDeniedException" is returned when the service is started.

Analysis: You are not authorized to access the configuration file.

Solution: Set the access permission of the configuration file to public read, or log on to Linux as an administrator to start the service.

- Task Counts are displayed as 0, but JobState is displayed as succeed.

Description: Pending Task Count and Diamond Task Count are both 0, but JobState is SUCCEED.

```
[2015-12-28 16:12:35] [INFO]  JobName:dir_data
[2015-12-28 16:12:35] [INFO]  Pending Task Count:0
[2015-12-28 16:12:35] [INFO]  Dispatched Task Count:0
[2015-12-28 16:12:35] [INFO]  Succeed Task Count:0
[2015-12-28 16:12:35] [INFO]  Failed Task Count:0
[2015-12-28 16:12:35] [INFO]  Is Scan Finished:true
[2015-12-28 16:12:35] [INFO]  JobState:SUCCEED
```

Analysis:

- Files cannot be listed because the value of `srcPrefix` is invalid.
- Folders specified by `srcPrefix` do not have files. Folders cannot be uploaded to OSS because the concept of folders is simulated by OSS.

Solution: Set a valid value for the `srcPrefix` parameter and make sure there are available files in the folder specified by `srcPrefix`. 
- "InvocationTargetException" is returned when you submit the job.

submit job:/disk2/ossimport2/local_job.cfg

Exception in thread "main" java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.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.simontuffs.Boot.run(OSSImport2.java:120)
... 6 more

Analysis: This error is returned because the configuration file or configuration file path is invalid.

Solution:

1. Configure the workingDir configuration item in the conf/sys.properties file properly.
2. If the configuration file is valid, check the configuration file path.

- Source files for synchronization do not exist.

Analysis: Master first lists the files and then migrates data based on the list. If files are deleted from the source after listing, they cannot be migrated. The deleted files are skipped during migration, and the names are listed in the error list.

- The running status of the migration job is inconsistent with the configuration file.

Description: The job configuration file is valid, but the job running status is inconsistent with the job configuration file during the migration.

Analysis: If a migration job has been submitted, modifications to the job configuration file do not take effect after the job is suspended.

Solution: Clear the previous job by using the clean command. After the job configuration file is modified, submit the job again.
Common problems about migrating data from UPYUN Storage Service (USS) to OSS

- **The number of jobs is always 0.**

  Analysis: View the job running log first.

  ```
  [2016-07-21 10:21:46] [INFO] [name=YoupaiList, totalRequest=1729925,
  avgLatency=38,
  recentLatency=300000]
  ```

  - If the value of `recentLatency` in the running log is smaller than or equal to
    30000, the files are listed properly. It can take more than 30 seconds (timeout)
    for files to be listed on USS in most cases. Files listed within 30 seconds are
    returned. This problem disappears when all files are listed.

  - If the value of `recentLatency` is small, this problem occurs because the
    account password is invalid. USS only returns null instead of error results
    when SDK errors occur. You can obtain the error code returned by USS for
    troubleshooting only by capturing packets.

- **How to specify `srcAccessKey` and `srcSecretKey`.**

  Enter the operator account and password of USS.

- **HTTP status code 429 keeps appearing.**

  USS has specified the SDK access interval. If too many requests are sent over a
  short period of time, the system limits the number of requests to be handled. We
  recommend that you contact the customer service of USS to remove the restrictions. ossimport retries in this case.

Common problems

- **After the migration job is complete, the data volume displayed in the OSS console is smaller than the source data volume.**

  Description: The bucket size does not change in the OSS console after all the
  migration job is complete, or the data sizes calculated by using the `du` command
  vary greatly in Linux.

  Analysis:

  - Bucket sizes in OSS console are updated with a delay of one to two hours.

  - The `du` command in Linux counts the block size, which is larger than the file
    size. You can use the `ls -lR <absolute folder path> | grep "\-rw"` command
    to list the file sizes.
'{{sum+=$5}END{{print sum}' command to count the actual size of the local folder.

- The error message such as unknown command "java" or unknown command "nohup" is returned when you run commands in Linux.

Analysis: The command that you want to run is not installed. Use the `yum`, `apt-get`, or `zypper` command to install the corresponding command based on operating systems.

- Can I use `srcPrefix` in the configuration file to specify files?

  You can only use `srcPrefix` to specify folders or prefixes.

- Can I configure proxies for `ossimport`?

  No. `ossimport` does not support proxies.

- Why am I charged for migrating data within OSS?

  If you migrate data by using an internal endpoint, you are charged based on the number of requests. No traffic fee is charged.

- Will files that are deleted from the local folder be deleted from OSS if the incremental data migration mode is enabled?

  No. If the incremental data migration mode is enabled, files that are deleted from the local folder will not be deleted from OSS.

- Why are there new files that are not synchronized when the incremental data migration mode is enabled?

  In incremental data migration mode, the last modification time of a file is used to determine whether the file is new. Linux `mv` command, Windows commands such as `cp`, `mv`, and `rsync with -t` or `-a` parameters do not modify the last modification time of files. Files modified by using these commands are not scanned for and therefore are ignored during migration.

- Can file permissions be migrated when files are migrated to OSS?

  No. After the migration is complete, you can use the `set-meta` command of `ossutil` to modify the permissions. For more information, see `set-meta`. 
5 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 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):

- ListObjects my-bucket
- ListObjects my-bucket/users
- ListObjects 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:
For more examples, see RAM Policy Editor README.
6 ossftp

6.1 Quick installation

ossftp is a FTP server tool based on Alibaba Cloud OSS. ossftp receives FTP requests and maps file and folder operations to OSS objects and buckets, allowing you to manage objects stored on OSS over FTP.

Features and functions

- **Features**
  - Cross-platform: You can run this tool from a graphical interface or using command lines on 32-bit or 64-bit Windows, Linux, and macOS operating systems.
  - Free of installation: You can run this tool directly after decompression.
  - Free of configuration: You can run this tool without further configurations.
  - Open source: The FTP tool is written in Python. You can view the complete source code. We will make the code available on GitHub.

- **Functions**
  - Upload, download, and delete objects and folders.
  - Supports multipart upload of large files.
  - Supports most FTP commands to meet everyday needs.

Note:

- ossftp is designed for individual users. We recommended that you use the OSS SDK for production environments.
- The ossftp V1.0 does not support Transport Layer Security (TLS) encryption for the ease of installation and deployment. FTP transmits data in plaintext. To prevent password leaks, we recommend that you run the FTP server and client on the same server and access using 127.0.0.1:port.
- Renaming and moving OSS objects through ossftp are not supported.
- The path to which the installation package is decompressed cannot contain Chinese characters.
The management control page of the FTP server may fail to open in Internet Explorer of earlier versions.

Python 2.6 and Python 2.7 are supported by FTP servers.

**Download link**

- **Windows**: ossftp-1.0.3-win.zip

  Python 2.7 is not installed on Windows by default. The installation package contains Python 2.7, and you can run ossftp directly after decompression.

- **Linux/macOS**: ossftp-1.0.3-linux-mac.zip

  Python 2.7 or Python 2.6 is installed on Linux or macOS by default. Therefore, the installation package only contains the required dependency libraries.

**Running**

1. Download the installation package based on your operating system.
2. Decompress the installation package and run it.
   - **Windows**
     
     Decompress the installation package and double-click `start.vbs`. If no response is returned, upgrade your Internet Explorer or set a different browser as the default browser.
   - **Linux**
     
     a. Decompress the installation package:
     
     ```bash
     $ unzip ossftp-1.0.3-linux-mac.zip
     
     b. Open the decompressed folder and run `start.sh`:
     
     ```bash
     $ cd ossftp-1.0.3-linux-mac
     $ bash start.sh
     
     c. Use a computer with a graphical interface to access the FTP server operation interface through a browser. Domain name: [http://ServerIP:8192](http://ServerIP:8192).
   - **macOS**
     
     Decompress the installation package and double-click `start.command`. Alternatively, run `bash start.command` on the terminal.
3. The preceding process starts an FTP server, which listens to port 2048 on IP address 127.0.0.1 by default. A web server is also started to listen to port 8192 on IP address 127.0.0.1 for the convenience of monitoring the FTP server status.

![Config](image)

- **ossftp address**: Set the IP address of the client that needs to use the FTP service. If the client is running on this server, you can keep the default setting.
- **ossftp port**: Set the listening port of ossftp. Keep the default setting.
- **ossftp log level**: Set the log level of ossftp. Set this parameter as needed.
- **Bucket endpoints**: Enter the domain name of the bucket in `bucket_name.endpoint` format. Separate multiple domain names with commas (,).
- **Language**: Set the display language for ossftp.

**Note:**

- After the configuration is complete, save the configuration and click Restart to apply the configuration.
- A server can only have a single connection at a time. When a new connection is created, the previous connection to the server is ended.

4. Download and install *FileZilla Client*. 
5. After configuring the OSS access information, you can click Quickconnect.

- Host: Set the server IP address. If the server and client are on the same device, use the default address 127.0.0.1.
- Username: Enter a bucket name and the AccessKey ID of the account that has the permission to access the bucket. The format is AccessKeyID/bucket_name. Example: tSxyi*****wPMEp/test-hz-jh-002.
- Password: Enter the AccessKey secret of the account that has the permission to access the bucket.
- Port: Enter the listening port configured on the server. The default value is 2048.

Note:
For more information about how to obtain the AccessKey ID and AccessKey secret, see the "Programmatic Access" section in Create a RAM user.

6.2 How to store remote attachments from Discuz! to OSS
This topic describes how to store remote attachments from Discuz! to OSS.

Prerequisites
- OSS is activated and a bucket whose ACL is Public Read is created.
  - For more information about how to activate OSS, see #unique_94.
  - For more information about how to create a bucket, see #unique_95.
- A Discuz! forum is built.

Context
The website remote attachment function allows you to directly store uploaded attachments to a remote storage server, which is usually a remote FTP server. Currently, Discuz! forums, PHPWind forums, and WordPress websites support the remote attachment function.
The version of Discuz! used for this topic is Discuz! X3.1.

Procedure

1. Log on to the Discuz! website with an administrator account.
2. On the management page, choose Global > Upload Settings.
3. Click the Remote Attachment tab and configure parameters.

<table>
<thead>
<tr>
<th>Configuration item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable remote attachment</td>
<td>Specifies whether to enable remote attachment. Select Yes.</td>
</tr>
<tr>
<td>Enable SSL connection</td>
<td>Specifies whether to enable SSL connection. Select No.</td>
</tr>
<tr>
<td>FTP server</td>
<td>Specifies the IP address that runs ossftp. We recommend that you set this parameter to 127.0.0.1.</td>
</tr>
<tr>
<td>FTP server port number</td>
<td>Specifies the port number of the FTP server. The default value is 2048.</td>
</tr>
<tr>
<td>FTP account</td>
<td>Specifies the FTP account in AccessKeyID/BucketName format. Note that the forward slash (/) is a delimiter and does not indicate an alternative.</td>
</tr>
<tr>
<td>FTP password</td>
<td>Specifies the FTP password AccessKey secret.</td>
</tr>
<tr>
<td>Passive mode connection</td>
<td>Specifies whether to enable passive mode connection. Select Yes.</td>
</tr>
<tr>
<td>Remote upload directory</td>
<td>Specifies the remote upload directory for attachments. We recommend that you set this parameter to a period (.) to create a directory for attachments in the root directory of the bucket.</td>
</tr>
<tr>
<td>Configuration item</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Remote access URL</td>
<td>Specifies the public endpoint of the bucket. The format is <a href="http://BucketName.Endpoint">http://BucketName.Endpoint</a>. In this example, the bucket resides in the China (Hangzhou) region, and the bucket name is test-hz-jh-002. Therefore, the URL is <a href="http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com">http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com</a>. For more information about endpoints, see #unique_96.</td>
</tr>
<tr>
<td>FTP timeout (seconds)</td>
<td>Specifies the FTP timeout period. Set the value to 0, indicating that the default setting of the server applies.</td>
</tr>
</tbody>
</table>

4. After the configuration is complete, click Test Remote Attachment to verify whether the configuration is correct.

5. Post a new article to verify whether the configuration is successful.
   a) Upload an image attachment to the article.
   b) Right-click the image and choose Open Link in New Tab from the shortcut menu.

   The URL in the following figure indicates that the image has been uploaded to the OSS bucket test-hz-jh-002.

6.3 How to store remote attachments from PHPWind to OSS

This topic describes how to store remote attachments from PHPWind to OSS.

Prerequisites

- OSS is activated and a bucket whose ACL is Public Read is created.
  - For more information about how to activate OSS, see #unique_94.
  - For more information about how to create a bucket, see #unique_95.
- A PHPWind forum is built.

Context

The website remote attachment function allows you to directly store uploaded attachments to a remote storage server, which is usually a remote FTP server.
Currently, Discuz! forums, PHPWind forums, and WordPress websites support the remote attachment function.

PHPWind8.7 is used for this topic.

Procedure

1. Log on to the PHPWind website with an administrator account.
3. Click the FTP Settings tab and configure parameters.

<table>
<thead>
<tr>
<th>Configuration item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable FTP uploads</td>
<td>Specifies whether to enable FTP uploads. Select Enable.</td>
</tr>
<tr>
<td>Website attachment URL</td>
<td>Specifies the public endpoint of the bucket. The format is <a href="http://BucketName.Endpoint">http://BucketName.Endpoint</a>. In this example, the bucket resides in the China (Hangzhou) region, and the bucket name is test-hz-jh-002. Therefore, the URL is <a href="http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com">http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com</a>. For more information about endpoints, see #unique_96.</td>
</tr>
<tr>
<td>FTP server</td>
<td>Specifies the IP address that runs ossftp. We recommend that you set this parameter to 127.0.0.1.</td>
</tr>
<tr>
<td>FTP server port number</td>
<td>Specifies the port number of the FTP server. The default value is 2048.</td>
</tr>
<tr>
<td>Remote upload directory</td>
<td>Specifies the remote upload directory for attachments. We recommend that you set this parameter to a period (.) to create a directory for attachments in the root directory of the bucket.</td>
</tr>
<tr>
<td>FTP account</td>
<td>Specifies the FTP account in AccessKeyID/BucketName format. Note that the forward slash (/) is a delimiter and does not indicate an alternative.</td>
</tr>
<tr>
<td>FTP password</td>
<td>Specifies the FTP password AccessKey secret.</td>
</tr>
<tr>
<td>FTP timeout (seconds)</td>
<td>Specifies the FTP timeout period. Set the value to 10. If no result is returned within 10 seconds, the system returns a timeout response.</td>
</tr>
</tbody>
</table>
4. Post a new article to verify whether the configuration is successful.
   a) Upload an image attachment to the article.
   b) Right-click the image and choose Open Link in New Tab from the shortcut menu.

   The URL in the following figure indicates that the image has been uploaded to the OSS bucket test-hz-jh-002.

6.4 How to store remote attachments from WordPress websites to OSS

This topic describes how to store remote attachments from WordPress websites to OSS.

Prerequisites

- OSS is activated and a bucket whose ACL is Public Read is created.
  - For more information about how to activate OSS, see #unique_94.
  - For more information about how to create a bucket, see #unique_95.
- A WordPress website is built.

Context

The website remote attachment function allows you to directly store uploaded attachments to a remote storage server, which is usually a remote FTP server. Currently, Discuz! forums, PHPWind forums, and WordPress websites support the remote attachment function.

WordPress does not provide native support for this function, but you can use a third-party plug-in to implement this function. In this topic, the WordPress version is 4.3.1 and the plug-in is Hacklog Remote Attachment.

Procedure

1. Log on to the WordPress website with an administrator account.
2. Click Plugin. Enter FTP in the Keyword search bar. Press Enter.
3. Find Hacklog Remote Attachment. Click Install Now.
4. After the plug-in is installed, choose Settings > Hacklog Remote Attachment.
5. In the Hacklog Remote Attachment dialog box that appears, configure the FTP service information.

<table>
<thead>
<tr>
<th>Configuration item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP server</td>
<td>Specifies the IP address that runs ossftp. We recommend that you set this parameter to 127.0.0.1.</td>
</tr>
<tr>
<td>FTP server port number</td>
<td>Specifies the port number of the FTP server. The default value is 2048.</td>
</tr>
<tr>
<td>FTP account</td>
<td>Specifies the FTP account in AccessKeyId/BucketName format. Note that the forward slash (/) is a delimiter and does not indicate an alternative.</td>
</tr>
<tr>
<td>FTP password</td>
<td>Specifies the FTP password AccessKey secret.</td>
</tr>
<tr>
<td>FTP timeout</td>
<td>Specifies the FTP timeout period. The default value is 30 seconds.</td>
</tr>
<tr>
<td>Remote basic URL</td>
<td>Specifies the public endpoint of the bucket. The format is <a href="http://BucketName.Endpoint">http://BucketName.Endpoint</a>. In this example, the bucket resides in the China (Hangzhou) region, and the bucket name is test-hz-jh-002. Therefore, the remote basic URL is <a href="http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/wp">http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/wp</a>. For more information about endpoints, see #unique_96.</td>
</tr>
<tr>
<td>FTP remote path</td>
<td>Specifies the path for storing the attachments in the bucket. In this example, wp indicates that all attachments are stored in the wp directory of the bucket. The Remote basic URL must correspond to the FTP remote path.</td>
</tr>
<tr>
<td>HTTP remote path</td>
<td>Specifies the HTTP remote path. We recommend that you set this parameter to a period (.).</td>
</tr>
</tbody>
</table>

6. Click Save.

Clicking Save triggers a test on the configuration. The test result is shown at the top of the page.
7. Post a new article to verify whether the configuration is successful.
   a) After you have written a new article, click Add Media to upload an attachment.
   b) Click Post. You can see the article that you have just written.
   c) Right-click the image and choose Open Link in New Tab from the shortcut menu.

   The URL in the following figure indicates that the image has been uploaded to the OSS bucket test-hz-jh-002.

6.5 FAQ

This topic describes the possible problems you may encounter when you use ossftp and their solutions.

- Failed to connect to the FTP server

  Possible causes:

  - The AccessKey ID and AccessKey secret are incorrect.

    Solution: Enter the correct information and try again.

  - The AccessKey information is the AccessKey pair of the RAM user account.

    The RAM user account is not authorized to list buckets.

    Solution: When you use a RAM user account for access, specify the bucket endpoint on the control page for the FTP server. At the same time, the RAM user account also needs some necessary permissions. For more information about how to use RAM to control access to OSS, see *Overview*. Details:

    ■ Read-only access

    To obtain the read-only permission when using ossftp, RAM users must obtain permissions to perform ListObjects, GetObject, and HeadObject. For more information about how to configure RAM policy, see *unique_101*.

    ■ Upload objects

    To upload objects, RAM users must obtain the permission to perform PutObject.

    ■ Delete objects

    To delete objects, RAM users must obtain the permission to perform DeleteObject.
• 501 and the corresponding error message are returned when you use FileZilla to transfer data after you run the FTP server in Linux

501 can't decode path (server filesystem encoding is ANSI_X3.4-1968)

The possible cause is that the Chinese character is not properly encoded. Run the following command on the terminal that runs start.sh and restart the terminal:

```
$ export LC_ALL=en_US.UTF-8; export LANG="en_US.UTF-8"; locale
```

• Disconnections due to timeout that occurs when you list objects after you log on to ossftp

The possible cause is that the root directory of the bucket contains an excessive number of files or folders. After you log on to ossftp, the FTP server tries to list all objects and folders in the root directory of the bucket. A maximum of 1,000 objects and folders can be listed each time. If the root directory contains more than 1 million objects and folders, more than 1,000 HTTP requests are sent, which causes timeout.

• Data fails to be transferred due to the port limit of the machine that runs the FTP server

The control port and data port for FTP are different. When the FTP server needs to transfer data in passive mode, a port is randomly selected and enabled for connections to the client. When the machine where the FTP server resides has port limits, data may fail to be transferred.

Solution: When ftpserver.py runs, specify the `--passive_ports_start` and `--passive_ports_end` parameters to set the range of the local port and enable the ports that fall within the range.

• Frequent disconnections between the client and FTP server

Each FTP client has timeout configurations. You can disable timeout configurations or set the timeout value to 0. Take FileZilla as an example. Choose Settings > Connection. Set the timeout value to 0.
7 ossfs

7.1 Quick installation

ossfs allows you to attach Alibaba Cloud OSS buckets to local files in Linux. In the system, you can use the local file system to perform operations on OSS objects to share data.

Functions

ossfs is built based on S3FS and incorporates all S3FS functions, including:

- Supports most functions of the POSIX-compliant file system, including file reading/writing, directories, link operations, permissions, UID/GID, and extended attributes.
- Uploads large objects by using OSS multipart upload.
- Supports MD5 verification which ensures data integrity.

Note:

You can use Cloud Storage Gateway (CSG) to access OSS. This way, OSS buckets are mapped to local directories or disks.

- CSG supports NFS and SMB (CIFS), allowing shared access to OSS-based directories.
- CSG also supports iSCSI. A large number of OSS buckets can be mapped to local disks to provide an efficient elastic storage solution.

Limits

Note the following limits before you use ossfs:

- Archive buckets cannot be attached by using ossfs.
- If you edit an uploaded object, the object is uploaded again.
- The performance of metadata-related operations such as `list directory` will degrade 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 data inconsistencies.
ossfs is not suitable for scenarios where read and write operations are highly concurrent.

Data consistency must be maintained when an OSS bucket is attached to multiple clients at the same time. 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.

Download URLs

<table>
<thead>
<tr>
<th>Released Linux</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubuntu 18.04 (x64)</td>
<td>ossfs_1.80.6_ubuntu18.04_amd64.deb</td>
</tr>
<tr>
<td>Ubuntu 16.04 (x64)</td>
<td>ossfs_1.80.6_ubuntu16.04_amd64.deb</td>
</tr>
<tr>
<td>Ubuntu 14.04 (x64)</td>
<td>ossfs_1.80.6_ubuntu14.04_amd64.deb</td>
</tr>
<tr>
<td>CentOS 7.0 (x64)</td>
<td>ossfs_1.80.6_centos7.0_x86_64.rpm</td>
</tr>
<tr>
<td>CentOS 6.5 (x64)</td>
<td>ossfs_1.80.6_centos6.5_x86_64.rpm</td>
</tr>
</tbody>
</table>

Installation

ossfs is subject to errors such as disconnections when run on earlier Linux kernel versions. Therefore, we recommend that you upgrade the operating system to CentOS 7.0 or Ubuntu 14.04 or a later version.

1. Download the installation package from the official website.

   Run the following command to download CentOS 7.0 (x64):

   ```
   wget http://gosspublic.alicdn.com/ossfs/ossfs_1.80.6_centos7.0_x86_64.rpm
   ```

2. Install ossfs.

   • Run the following commands to install Ubuntu:

   ```
   sudo apt-get update
   sudo apt-get install gdebi-core
   sudo gdebi your_ossfs_package
   ```

   • Run the following command to install CentOS 6.5 or a later version:

   ```
   sudo yum localinstall your_ossfs_package
   ```

Note:
If your client uses yum to install the RPM package, dependent packages may fail to be downloaded when the client node network environment is not suitable. To resolve this issue, use yum to download dependent packages to the node that runs the same operating system version over the normal network. Copy the dependent package to the required node where the network environment is not suitable. For example, ossfs depends on fuse 2.8.4 or a later version. Run the following command to download the latest version of fuse from the yum source to your local device:

```
sudo yum install --downloadonly --downloaddir=./ fuse
```

To download other dependent packages, replace fuse with the name of the required package.

- Run the following command to install CentOS 5:

```
sudo yum localinstall your_ossfs_package --nogpgcheck
```

3. Configure access information of the account.

Store the bucket name, AccessKey IDs, and AccessKey secrets that can access the bucket in the `/etc/passwd-ossfs` object. Note that the permissions on this file must be set correctly. We recommend that you set the permission to 640.

```
echo my-bucket:my-access-key-id:my-access-key-secret > /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
```

4. Attach the bucket to the specified directory.

```
ossfs my-bucket my-mount-point -ourl=my-oss-endpoint
```

**Example: Attach a bucket named `bucket-test` to the `/tmp/ossfs` directory.**

```
echo bucket-test:LTAIbZcdVCmQ*****:M0k8x0y9hxQ31coh7A5e2MZEUz***** > /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
mkdir /tmp/ossfs
ossfs bucket-test /tmp/ossfs -ourl=http://oss-cn-hangzhou.aliyuncs.com
```

**Notice:**

If you use ECS purchased from Alibaba Cloud to provide ossfs services, you can use an internal endpoint. In the preceding example, you can set the OSS endpoint to `oss-cn-hangzhou-internal.aliyuncs.com` to minimize costs. For more information about OSS internal endpoints, see Regions and endpoints.
5. Detach the bucket if you no longer need to attach this bucket.

```
fusermount -u /tmp/ossfs
```

**Note:**
For more information, visit [GitHub ossfs](https://github.com).

**Version logs**
For more information about version logs, visit [GitHub ChangeLog](https://github.com).

### 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 \(-d\) \(-o\) \(\text{f2}\) 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 \(-f\) \(-d\) \(-o\) \(\text{f2}\) 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_other parameter as follows when running ossfs:

  ```bash
  ossfs your_bucket your_mount_point -ourl=your_endpoint -o allow_other
  ```

- 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_mountpoint -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 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_mountpoint -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 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: 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:

```bash
mkdir /root/ossfs_scripts
vi /root/ossfs_scripts/start_ossfs.sh
```

Write the following data:

```bash
# 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_scripts/start_ossfs.sh
logfile=/var/log/ossfs.log
log_stdout=true
log_stderr=true
logfile_maxbytes=1MB
logfile_backups=10
```

4. Run Supervisor:

```
supervisord
```

5. 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 /osscmd, 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 parameter:

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.

```bash
$ 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:

```bash
$ 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:

```bash
$ python osscmd gs --id=your_id --key=your_key --host=your_endpoint
2013-07-19 08:11 test-oss-sample
Bucket Number is: 1
```

To configure the AccessKey ID and AccessKey Secret and import them to the default file, run the following command. The default domain name of OSS is oss.aliyuncs.com.

```bash
$python osscmd config --id=your_id --key=your_key --host=your_endpoint
```

If a similar output Your configuration is saved into is displayed, the AccessKey ID and AccessKey Secret have been saved.

Basic operations

- List created buckets

```bash
$python osscmd getallbucket
```

No buckets are displayed if the user has no buckets in OSS.

- Create a bucket

Create a bucket named mybucketname.

```bash
$python osscmd createbucket mybucketname
```

The bucket may fail to be created. The cause is that the name of a bucket must be unique in OSS. You need to use another bucket name. For example, you can add a specific date to the bucket name.
- Check whether a bucket is created

```python
$python osscmd getallbucket
```

If a bucket fails to be created, check the error message returned from the `osscmd` tool.

- View objects

After a bucket is created, you can view the objects in the bucket.

```python
$python osscmd list oss://mybucketname/
```

If there are no objects in the bucket, no objects are displayed.

- Upload a file

Upload a local file to a bucket. If a local file is named `local_existed_file`, run the following commands to calculate the MD5 value of the file and upload the local file to the bucket:

```bash
$ md5sum local_existed_file 7625e1adc3a4b129763d580ca0a78e44
local_existed_file
$ python osscmd put local_existed_file oss://mybucketname/test_object
```

**Note:**

`md5sum` runs in Linux only.

- View objects again

You can view objects that have been uploaded to a bucket.

```python
$python osscmd list oss://mybucketname/
```

- Download an object

Download an object from a bucket to a local file. Compare the MD5 value of the object that is downloaded with the object MD5 value calculated before the object is downloaded.

```bash
$ python osscmd get oss://mybucketname/test_object download_file
$ md5sum download_file 7625e1adc3a4b129763d580ca0a78e44 download_file
```

**Note:**

`md5sum` runs in Linux only.
Object Storage Service

- 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.

  ```xml
  <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 the [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
  <? 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>
  ```
**Delete a lifecycle rule.**

```python
python osscmd deletelifecycle oss://mybucket
```

0.139(s) elapsed

**Read a lifecycle rule.**

```python
python osscmd getlifecycle oss://mybucket
```

Error Headers:
```
[('content-length', '288'), ('server', 'AliyunOSS'), ('connection', 'close'), ('x-oss-request-id', '54C74FEE5D7F6B24E5042630'), ('date', 'Tue, 27 Jan 2015 08:44:30 GMT'), ('content-type', 'application/xml')]
```

Error Body:
```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <BucketName>mybucket</BucketName>
  <Code>NoSuchLifecycle</Code>
  <Message>No Row found in Lifecycle Table.</Message>
  <RequestId>54C74FEE5D7F6B24E5042630</RequestId>
  <HostId>mybucket.oss-maque-hz-a.alibaba.net</HostId>
</Error>
```

Error Status:
404

getlifecycle Failed!

Configure hotlinking protection

- **Allow access from a request that has an empty Referer field.**

  ```bash
  $osscmd putreferer oss://test --allow_empty_referer=true
  0.004(s) elapsed
  ```

- **Obtain the Referer whitelist.**

  ```bash
  $osscmd getreferer oss://test
  <? xml version="1.0" encoding="UTF-8"?>
  <RefererConfiguration>
    <AllowEmptyReferer>false</AllowEmptyReferer>
    <RefererList>
      <Referer>www.test.com</Referer>
    </RefererList>
  </RefererConfiguration>
  ```

- **The Referer field is required. Only requests that have the Referer field value of test are allowed.**

  ```bash
  $osscmd putreferer oss://test --allow_empty_referer=false --referer='www.test.com'
  0.092(s) elapsed
  ```

- **Obtain the Referer whitelist.**

  ```bash
  $osscmd getreferer oss://test
  <? xml version="1.0" encoding="UTF-8"?>
  <RefererConfiguration>
    <AllowEmptyReferer>true</AllowEmptyReferer>
    <RefererList />
  </RefererConfiguration>
  ```
• The Referer field is required. Only requests that have the Referer field value of test or test1 are allowed.

```bash
```

• Obtain the Referer whitelist.

```bash
$osscmd getreferer oss://test
<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
  <AllowEmptyReferer>false</AllowEmptyReferer>
  <RefererList>
    <Referer>www.test.com</Referer>
    <Referer>www.test1.com</Referer>
  </RefererList>
</RefererConfiguration>
```

Use logging

• Configure a logging rule

```bash
$osscmd putlogging oss://mybucket oss://myloggingbucket/mb
```

• Obtain logging rules that are configured for a bucket

```bash
$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:

```bash
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
getallbucket(gs)

Command:

getallbucket(gs)

Obtain created buckets. gs is short for get allbucket. You can run the gs or allbucket command to obtain a list of created buckets.

Example:

• python osscmd getallbucket
• python osscmd gs

createbucket(cb,mb,pb)

Command:

createbucket(cb,mb,pb) oss://bucket --acl=[acl]

Create a bucket.

• cb is short for create bucket. mb is short for make bucket. pb is short for put bucket.
• You can set oss://bucket to specify a bucket name.
• The acl parameter is optional.

Example:

• python osscmd createbucket oss://mybucket
• python osscmd cb oss://myfirstbucket --acl=public-read
• python osscmd mb oss://mysecondbucket --acl=private
• python osscmd pb oss://mythirdbucket

deletebucket(db)

Command:

deletebucket(db) oss://bucket

Delete a bucket. db is short for delete bucket.

Example:

• python osscmd deletebucket oss://mybucket
python osscmd db oss://myfirstbucket

deletewholebucket

**Warning:**
All data is deleted if you run this command. Deleted data cannot be recovered.
Exercise caution when you run this command.

**Command:**

deletewholebucket oss://bucket

Delete a bucket, and all objects and fragments in the bucket.

**Example:**

```bash
python osscmd deletewholebucket oss://mybucket
```

getacl

**Command:**

getacl oss://bucket

Obtain the bucket ACL.

**Example:**

```bash
python osscmd getacl oss://mybucket
```

setacl

**Command:**

```bash
setacl oss://bucket --acl=[acl]
```

Modify the bucket ACL. You can set the bucket ACL to private, public-read, or public-read-write.

**Example:**

```bash
python osscmd setacl oss://mybucket --acl=private
```

putlifecycle

**Command:**

```bash
putlifecycle oss://mybucket lifecycle.xml
```
Set lifecycle rules. In the command, lifecycle.xml indicates a file that is used to configure lifecycle rules. For more information, see API Reference.

Example:

```bash
python osscmd putlifecycle oss://mybucket lifecycle.xml
```

```xml
<LifecycleConfiguration>
  <Rule>
    <ID>1125</ID>
    <Prefix>log_backup/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>2</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

getchifecycle

**Command:**

```bash
osscmd getlifecycle oss://bucket
```

Obtain lifecycle rules of a bucket.

**Example:**

```bash
python osscmd getlifecycle oss://mybucket
```

deletelifecycle

**Command:**

```bash
osscmd deletelifecycle oss://bucket
```

Delete all lifecycle rules of a bucket.

**Example:**

```bash
python osscmd deletelifecycle oss://mybucket
```

putreferer

**Command:**

```bash
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:

```
```

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:**

```bash
mkdir oss://bucket/dname
```

**Create a folder.**

**Example:**

```bash
python osscmd mkdir oss://mybucket/folder
```

listallobject

**Command:**

```bash
listallobject oss://bucket/[prefix]
```

**List all objects in a bucket. You can specify a prefix to list objects whose names start with the prefix.**

**Example:**

- `python osscmd listallobject oss://mybucket`
- `python osscmd listallobject oss://mybucket/testfolder/`

deleteallobject

**Command:**

```bash
deleteallobject oss://bucket/[prefix]
```

**Delete all objects in a bucket. You can also specify a prefix to delete objects whose names start with the prefix.**

**Example:**

- `python osscmd deleteallobject oss://mybucket`
- `python osscmd deleteallobject oss://mybucket/testfolder/`

downloadallobject

**Command:**

```bash
downloadallobject oss://bucket/[prefix] localdir --replace=false --thread_num=5
```

**Download objects from a bucket to a local directory. This operation ensures that the original directory structure remains the same. You can specify a prefix to**
download objects whose names start with the specified prefix. —replace=false indicates that local files with the same name of the object will not be overwritten during the download. —replace=true indicates that local files with the same name of the object will be overwritten. You can also use thread_num to configure the download thread.

Example:

- python osscmd downloadallobject oss://mybucket /tmp/folder
- python osscmd downloadallobject oss://mybucket /tmp/folder --replace=false
- python osscmd downloadallobject oss://mybucket /tmp/folder --replace=true --thread_num=5

downloadtodir

Command:

downloadtodir oss://bucket/[prefix] localdir --replace=false

Download objects from a bucket to a local directory. This operation ensures that the original directory structure remains the same. You can specify a prefix to download objects whose names start with the specified prefix. —replace=false indicates that local files with the same name of the object will not be overwritten during the download. —replace=true indicates that local files with the same name of the object will be overwritten. downloadtodir follows the same logic as that of downloadallobject.

Example:

- python osscmd downloadtodir oss://mybucket /tmp/folder
- python osscmd downloadtodir oss://mybucket /tmp/folder --replace=false
- python osscmd downloadtodir oss://mybucket /tmp/folder --replace=true

uploadfromdir

Command:

uploadfromdir localdir oss://bucket/[prefix] --check_point=check_point_file --replace=false --check_md5=false --thread_num=5

Upload local files to a bucket.
If local directory `/tmp/` contains the `a/b`, `a/c`, and `a` files, the paths of these files in OSS are `oss://bucket/a/b`, `oss://bucket/a/c`, and `oss://bucket/a`. If a prefix is set to `mytest`, the paths of these files in OSS are `oss://bucket/mytest/a/b`, `oss://bucket/mytest/a/c`, and `oss://bucket/mytest/a`.

`--check_point=check_point_file` is used to specify a checkpoint file. After the checkpoint file is specified, the `osscmd` tool will be used to store the timestamps that are recorded when the local files are uploaded. The `uploadfromdir` command is used to compare the timestamps of the files that are being uploaded and the timestamps that are recorded in the checkpoint file. If the timestamps are different, the files are reuploaded. `check_point_file` is not specified by default. `--replace=false` indicates that local files with the same name of the object will not be overwritten during the upload. `--replace=true` indicates that local files with the same name of the object will be overwritten. `--check_md5=false` indicates that Content-MD5 is not included in the request header and MD5 verification will not be performed. `--check_md5=true` indicates that MD5 verification will be performed.

Note: The checkpoint file stores upload records of all objects.

Example:

- `python osscmd uploadfromdir /mytemp/folder oss://mybucket`
- `python osscmd uploadfromdir /mytemp/folder oss://mybucket --check_point_file=/tmp/mytemp_record.txt`
- `python osscmd uploadfromdir C:\Documents and Settings\User\My Documents\Downloads oss://mybucket --check_point_file=C:\cp.txt`

Command:

```
put localfile oss://bucket/object --content-type=[content_type] --headers="key1:value1#key2:value2" --check_md5=false
```

When uploading a local file to a bucket, you can set HTTP header fields such as `content-type`. `--check_md5=false` indicates that Content-MD5 is not included in the request header and MD5 verification will not be performed. `--check_md5=true` indicates that MD5 verification will be performed.
Example:

- python osscmd put myfile.txt oss://mybucket
- python osscmd put myfile.txt oss://mybucket/myobject.txt
- python osscmd put myfile.txt oss://mybucket/test.txt --content-type=plain/text
  --headers="x-oss-meta-des:test#x-oss-meta-location:CN"
- python osscmd put myfile.txt oss://mybucket/test.txt --content-type=plain/text

upload

Command:

upload localfile oss://bucket/object --content-type=[content_type]
  --check_md5=false

Upload local files to a bucket. --check_md5=false indicates that Content-MD5 is not included in the request header and MD5 verification will not be performed. --check_md5=true indicates that MD5 verification will be performed.

Example:

python osscmd upload myfile.txt oss://mybucket/test.txt
  --content-type=plain/text

get

Command:

get oss://bucket/object localfile

Download an object to a local file.

Example:

python osscmd get oss://mybucket/myobject /tmp/localfile

multiget(multi_get)

Command:

multiget(multi_get) oss://bucket/object localfile --thread_num=5

Use multithreading to download an object to a local file. You can configure the number of threads that are used to download the object.

Example:
python osscmd multiget oss://mybucket/myobject /tmp/localfile
python osscmd multi_get oss://mybucket/myobject /tmp/localfile

**cat**

**Command:**

cat oss://bucket/object

Read and display object content. Do not run this command if the object is large.

**Example:**

python osscmd cat oss://mybucket/myobject

**meta**

**Command:**

meta oss://bucket/object

Read and display the meta information of the object. Meta information contains the content-type, file length, and user metadata.

**Example:**

python osscmd meta oss://mybucket/myobject

**copy**

**Command:**

copy oss://source_bucket/source_object oss://target_bucket/target_object --headers="key1:value1#key2:value2"

Replicate an object from a source bucket to a destination bucket.

**Example:**

python osscmd copy oss://bucket1/object1 oss://bucket2/object2

**rm(delete,del)**

**Command:**

rm(delete,del) oss://bucket/object --encoding_type=url

Delete an object. When encoding-type is set to url, control characters to be deleted also need to be URL-encoded.
Example:

- python osscmd rm oss://mybucket/myobject
- python osscmd delete oss://mybucket/myobject
- python osscmd del oss://mybucket/myobject
- python osscmd del oss://mybucket/my%01object --encoding_type=url

signurl(sign)

Command:

signurl(sign) oss://bucket/object --timeout=[timeout_seconds]

Generate a signed URL containing the timeout value. A signed URL is used to provide access to a specific object when the bucket ACL is private.

Example:

- python osscmd sign oss://mybucket/myobject
- python osscmd signurl oss://mybucket/myobject

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=75835E389EA648C0B93571B6A46023F3
```

listparts

Command:

listparts oss://bucket

List the objects and upload IDs of multipart upload events that have not been completed for a bucket. When you want to delete a bucket but the system prompts that the bucket is not empty, you can run this command to check whether there are fragments in the bucket.

Example:

```
python osscmd listparts oss://mybucket
```

getallpartsize

Command:

getallpartsize oss://bucket

List the total size of parts that are uploaded by using the existing upload IDs.

Example:

```
python osscmd getallpartsize oss://mybucket
```
cancel

Command:

cancel oss://bucket/object --upload_id=xxx

Terminate the multipart upload event that uses the upload ID.

Example:

```
python osscmd cancel oss://mybucket/myobject --upload_id=
```
multiupload(multi_upload, mp)

**Command:**

```
multiupload(multi_upload, mp) localfile oss://bucket/object --check_md5=false --thread_num=10
```

Use multipart upload to upload a local file to OSS.

**Example:**

- python osscmd multiupload /tmp/localfile.txt oss://mybucket/object
- python osscmd multiup_load /tmp/localfile.txt oss://mybucket/object
- python osscmd mp /tmp/localfile.txt oss://mybucket/object

**Command:**

```
multiupload(multi_upload, mp) localfile oss://bucket/object --upload_id=xxx --thread_num=10 --max_part_num=1000 --check_md5=false
```

Use multipart upload to upload a local file to OSS. The part count of the local file is defined by the max_part_num parameter. When this command is run, the system first determines whether the MD5 value of ETags of parts that use the upload ID is the same with the MD5 value of the local file. If their values are the same, the parts are uploaded. Generate an upload ID before this upload event is started. Add the upload ID to the command. If the upload fails, you can run the same multiupload command to upload the parts in the same way you use resumable upload. --check_md5=false indicates that Content-MD5 is not included in the request header and MD5 verification will not be performed. --check_md5=true indicates that MD5 verification will be performed.

**Example:**

- python osscmd multiupload /tmp/localfile.txt oss://mybucket/object --upload_id=D9D278DB6F8845E9AFE797DD235DC576
- python osscmd multiup_load /tmp/localfile.txt oss://mybucket/object --thread_num=5
- python osscmd mp /tmp/localfile.txt oss://mybucket/object --max_part_num=100
copylargefile

Command:

```bash
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:

```bash
python osscmd copylargefile oss://source_bucket/source_object
    oss://target_bucket/target_object --part_size=10*1024*1024
```

uploadpartfromfile (upff)

Command:

```bash
uploadpartfromfile (upff) localfile oss://bucket/object --upload_id=xxx
   --part_number=xxx
```

This command is used for tests only.

uploadpartfromstring(upfs)

Command:

```bash
uploadpartfromstring(upfs) oss://bucket/object --upload_id=xxx
   --part_number=xxx
   --data=xxx
```

This command is used for tests only.