

Alibaba Cloud

对象存储 OSS
常用工具

文档版本：20220712

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.OSS常用工具汇总	08
2.命令行工具ossutil	10
2.1. 概述	10
2.2. 下载和安装	11
2.3. 常用命令	14
2.3.1. appendfromfile（追加上传）	14
2.3.2. bucket-cname（自定义域名）	15
2.3.3. bucket-encryption（服务器端加密）	18
2.3.4. bucket-policy（授权策略）	20
2.3.5. bucket-tagging（存储空间标签）	26
2.3.6. bucket-versioning（版本控制）	27
2.3.7. cat（输出文件内容）	29
2.3.8. config（创建配置文件）	31
2.3.9. cors（跨域资源共享）	33
2.3.10. cors-options（检测跨域请求）	35
2.3.11. cp（上传、下载和拷贝文件）	37
2.3.11.1. 简介	37
2.3.11.2. 上传文件	37
2.3.11.3. 下载文件	44
2.3.11.4. 拷贝文件	50
2.3.12. create-symlink（创建软链接）	56
2.3.13. du（获取大小）	57
2.3.14. getallpartsize（获取碎片大小）	59
2.3.15. hash（计算CRC64或MD5）	60
2.3.16. help（获取帮助信息）	61
2.3.17. inventory（清单）	62

2.3.18. lcb (列举云盒信息)	65
2.3.19. lifecycle (生命周期)	66
2.3.20. listpart (列举碎片)	70
2.3.21. logging (日志转存)	71
2.3.22. lrb (列举地域级别下的Bucket)	74
2.3.23. ls (列举账号级别下的资源)	75
2.3.24. mb (创建存储空间)	82
2.3.25. mkdir (创建目录)	84
2.3.26. object-tagging (对象标签)	85
2.3.27. probe (探测状态)	88
2.3.28. read-symlink (读取软链接)	96
2.3.29. referer (防盗链)	98
2.3.30. replication (跨区域复制)	100
2.3.31. request-payment (请求者付费)	104
2.3.32. restore (解冻文件)	106
2.3.33. revert-versioning (恢复版本)	110
2.3.34. rm (删除)	112
2.3.35. set-acl (设置或修改ACL)	116
2.3.36. set-meta (管理文件元信息)	118
2.3.37. sign (生成签名URL)	123
2.3.38. stat (查看Bucket和Object信息)	125
2.3.39. sync (同步文件)	127
2.3.39.1. 简介	127
2.3.39.2. 同步本地文件到OSS	128
2.3.39.3. 同步OSS文件到本地	132
2.3.39.4. 在OSS之间同步文件	136
2.3.40. update (ossutil版本升级)	141
2.3.41. website (静态网站托管及回源配置)	141

2.3.42. worm（合规保留策略）	146
2.4. 查看选项	148
2.5. 常见问题	155
3. 图形化管理工具ossbrowser	158
3.1. 安装并登录ossbrowser	158
3.2. 快速使用ossbrowser	160
3.3. 权限管理	161
3.4. 常见问题	164
4. 数据迁移工具ossimport	167
4.1. 说明及配置	167
4.2. 单机部署	176
4.3. 分布式部署	180
4.4. 常见问题	183
5. ossfs	191
5.1. 概述	191
5.2. 快速安装	191
5.3. 高级配置	193
5.4. 查看选项	197
5.5. 常见问题	200
6. ossftp	206
6.1. 概述	206
6.2. 安装ossftp	207
6.3. 创建自定义登录用户	209
6.4. 使用案例	211
6.4.1. FileZilla如何上传本地站点文件到OSS	211
6.4.2. Discuz如何存储远程附件到OSS	212
6.4.3. Phpwind如何存储远程附件到OSS	213
6.4.4. WordPress如何存储远程附件到OSS	214

6.5. 常见问题	216
7. osscmd (已下线)	218
7.1. 快速安装	218
7.2. 使用示例	218
7.3. 有关Bucket命令	222
7.4. 有关Object命令	226
7.5. 有关Multipart命令	231
8. RAM策略编辑器	234

1.OSS常用工具汇总

您可以通过以下常用工具更高效地使用OSS。

工具	简介
ossbrowser	图形化的管理工具。 • 图形化界面，使用简单。 • 提供类似Windows资源管理器的功能。 • 支持直接浏览文件。 • 支持文件目录（文件夹）的上传下载。 • 支持文件并发上传、断点续传。 • 支持RAM 用户的图形化Policy授权操作。 • 支持Windows、Linux、Mac平台。 使用限制： • ossbrowser是图形化工具，传输速度和性能不如ossutil。 • 只支持5 GB以下的文件移动或复制。 • 单文件上传最大不能超过48.8 TB。
ossutil	Object和Bucket的命令行管理工具。 • 提供方便、简洁、丰富的Object和Bucket管理命令，操作性能好。 • 支持文件并发上传、断点续传。 • 支持文件目录（文件夹）的上传下载。
osscommand (已下线)	Object和Bucket的命令行管理工具。 • 提供完备的Bucket、Object管理命令。 • 支持Windows、Linux平台。 使用限制： • 仅适用于Python2.5~2.7版本，不支持Python 3.x版本。 • 不支持低频访问、归档存储、冷归档存储、跨区域复制、镜像回源等功能。  注意 osscommand操作命令已整合到ossutil中。osscommand已于2019年07月31日下线，给您带来不便敬请谅解。

工具	简介
ossfs	Bucket挂载工具。ossfs用于将OSS的Bucket挂载到Linux系统的本地文件系统中，挂载后可通过本地文件系统操作OSS上的Object，实现数据的访问和共享。 - 支持POSIX文件系统的大部分功能，包括文件读写、目录、链接操作、权限、UID、GID、以及扩展属性（Extended Attributes）。 - 支持使用OSS的Multipart功能上传大文件。 - 支持MD5校验，保证数据完整性。 使用限制： - 不支持挂载归档和冷归档类型Bucket。 - 编辑已上传文件会导致文件被重新上传。 - 元数据操作，例如 <code>list directory</code>，因为需要远程访问OSS服务器，所以性能较差。 - 重命名文件或文件夹可能会出错。若操作失败，可能会导致数据不一致。 - 不适合高并发读写的场景。 - 多个客户端挂载同一个OSS Bucket时，数据一致性由您自行维护。建议您合理规划文件使用时间，避免出现多个客户端写同一个文件的情况。 - 不支持Hard Link。
ossftp	管理Object的FTP工具。 - 使用FileZilla、WinSCP、FlashFXP等FTP客户端操作OSS。 - 本质是FTP Server，用于接收FTP请求，会将文件、文件夹的操作映射为对OSS的操作。 - 基于Python2.7及以上版本。 - 支持Windows、Linux、Mac平台。
ossimport	OSS数据同步工具。 - 可将各类第三方数据源文件同步到OSS上。 - 支持分布式部署，可使用多台服务器批量迁移数据。 - 支持TB级以上数据迁移。 - 支持Windows、Linux平台。 - 适用于Java 1.7及1.8版本。
RAM策略编辑器	OSS授权策略自动化生成工具。当您需要生成自定义授权策略时，推荐使用该工具。 - 可根据需求自动生成授权策略，还可以在RAM的自定义权限策略中使用该授权策略。 - 支持浏览器Chrome、Firefox、Safari。

2. 命令行工具ossutil

2.1. 概述

ossutil支持通过Windows、Linux和macOS系统以命令行方式管理OSS数据。

安装ossutil

有关ossutil的下载及安装的具体操作，请参见[下载和安装](#)。

常用命令

您可以通过以下列表学习ossutil命令的用法：

名称	描述
<code>appendfromfile</code>	用于在已上传的追加类型文件（Appendable Object）末尾直接追加内容。
<code>bucket-cname</code>	查看Bucket的CNAME配置。
<code>bucket-encryption</code>	添加、修改、查询、删除Bucket的加密配置。
<code>bucket-policy</code>	添加、修改、查询、删除Bucket的Bucket policy配置。
<code>bucket-tagging</code>	添加、修改、查询、删除Bucket的标签配置。
<code>bucket-versioning</code>	添加或查询Bucket的版本控制配置。
<code>cat</code>	将文件内容输出到ossutil。
<code>config</code>	创建配置文件来存储OSS访问信息。
<code>cors</code>	添加、修改、查询、删除Bucket的CORS配置。
<code>cors-options</code>	用于测试Bucket是否允许指定的跨域访问请求。
<code>cp简介</code>	用于上传、下载、拷贝文件。
<code>create-symlink</code>	创建符号链接（软链接）。
<code>du</code>	用于获取指定Bucket、指定Object或文件目录所占的存储空间大小。
<code>getallpartsize</code>	获取Bucket内所有未完成上传的Multipart任务的每个分片大小以及分片总大小。
<code>hash</code>	用于计算本地文件的CRC64或MD5。
<code>help</code>	获取命令的帮助信息。当您不清楚某个命令的用法时，建议您使用 <code>help</code> 命令获取该命令的帮助信息。
<code>inventory</code>	命令用于添加、查询、列举、删除Bucket的清单规则。
<code>lifecycle</code>	命令用于添加、修改、查询、删除生命周期规则配置。

名称	描述
listpart	列出没有完成分片上传的Object的分片信息。
logging	添加、修改、查询、删除Bucket的日志管理配置。
lrb	列举单个或多个地域（Region）下Bucket的基本信息。
ls	列举Bucket、Object和Part。
mb	创建Bucket。
mkdir	在Bucket内创建文件目录。
object-tagging	添加、修改、查询或删除Object的标签配置。
probe	针对OSS访问的检测命令，可用于排查上传、下载过程中因网络故障或基本参数设置错误导致的问题。
read-symlink	读取符号链接（软链接）文件的描述信息。
referer	添加、修改、查询、删除Bucket的防盗链配置。
replication	管理Bucket的跨区域复制规则配置。
request-payment	设置或查询Bucket的请求者付费模式配置。
restore	恢复冷冻状态的Object为可读状态。
revert-versioning	将已删除的Object恢复至最近的版本。
rm（删除）	删除Bucket、Object和Part。
set-acl	设置Bucket或Object的访问权限（ACL）。
set-meta	设置已上传Object的元信息。
sign	用于生成经过签名的文件URL，并将签名URL分享给第三方供其下载或预览。
stat	获取指定Bucket或Object的描述信息。
sync简介	用于同步本地文件到OSS、同步OSS文件到本地、在OSS之间同步文件。
update	用于更新ossutil版本。
website	用于添加、修改、查询、删除Bucket的静态网站托管配置、重定向配置、镜像回源配置。
worm	管理Bucket的合规保留策略。

2.2. 下载和安装

ossutil支持在Windows、Linux、macOS等系统中运行，您可以根据实际环境下载和安装合适的版本。

版本和运行环境

- 当前版本：1.7.13
- 软件源码：[ossutil](#)
- 运行环境
 - Windows/Linux/macOS
 - 支持架构：x86（32bit、64bit）、ARM（32bit、64bit）

下载地址

ossutil1.7.8版本基于Golang 1.16版本编译，不支持通过 macOS x86 32bit 操作系统使用ossutil。因此，ossutil1.7.8及以上版本均不再提供 ossutilmac32 安装包。

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

根据您的操作系统下载对应版本，然后按如下步骤安装并配置ossutil，以下均以64位系统为例。

Linux系统安装

1. 运行以下命令下载ossutil。

```
wget https://gosspublic.alicdn.com/ossutil/1.7.13/ossutil64
```

 **注意** 复制下载地址时会自动加上spm编码内容，使用wget命令下载时注意删除链接中 `?spm=xxxx` 的内容。

2. 运行以下命令修改文件执行权限。

```
chmod 755 ossutil64
```

3. 使用交互式配置生成配置文件。

- i. 输入配置命令。

```
./ossutil64 config
```

ii. 根据提示设置配置文件路径。

建议直接按回车使用默认配置文件的路径。

请输入配置文件名，文件名可以带路径（默认为：`/home/user/.ossutilconfig`，回车将使用默认路径。如果用户设置为其它路径，在使用命令时需要将`--config-file`选项设置为该路径）：

ossutil默认使用`/home/user/.ossutilconfig`作为配置文件，若您设置了配置文件的路径，则每次使用命令时需增加`-c`选项指定配置文件。例如配置文件保存为`/home/config`，使用ls时，命令格式如下：

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

iii. 根据提示设置工具的语言。

请输入语言`CH`或`EN`。工具使用的语言默认与操作系统保持一致。该配置项将在此次config命令设置成功后生效。

iv. 根据提示分别设置Endpoint、AccessKey ID、AccessKey Secret和STSToken参数。

参数说明如下：

- endpoint：填写Bucket所在地域的Endpoint。各地域Endpoint详情，请参见[访问域名和数据中心](#)。

您也可以增加 `http://` 或 `https://` 指定ossutil访问OSS使用的协议，默认使用HTTP协议。例如使用HTTPS协议访问深圳的Bucket，设置为 `https://oss-cn-shenzhen.aliyuncs.com`。

- accessKeyID、accessKeySecret：填写账号的AccessKey。
 - 使用阿里云账号或RAM用户访问时，AccessKey的获取方式，请参见[获取AccessKey](#)。
 - 使用STS临时授权账号访问时，AccessKey的获取方式，请参见[使用STS临时访问凭证访问OSS](#)。
- stsToken：使用STS临时授权账号访问OSS时需要配置该项，否则置空即可。关于stsToken的生成方式，请参见[临时访问凭证](#)。

 说明 关于配置文件的更多信息，请参见[config](#)。

Windows系统安装

1. 单击下载链接下载工具。
2. 将工具解压，并双击运行`ossutil.bat`文件。
3. 运行以下命令，生成配置文件。

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

4. 根据提示设置配置文件。设置方法与[Linux系统生成配置文件](#)的方法相同。

macOS系统安装

1. 运行以下命令下载ossutil。

```
curl -o ossutilmac64 https://gosspublic.alicdn.com/ossutil/1.7.13/ossutilmac64
```

2. 运行以下命令修改文件执行权限。

```
chmod 755 ossutilmac64
```

3. 运行以下命令，生成配置文件。

```
./ossutilmac64 config
```

4. 根据提示设置配置文件。设置方法与Linux系统生成配置文件的方法相同。

ARM系统安装

1. 运行以下命令下载ossutil。

```
wget https://gosspublic.alicdn.com/ossutil/1.7.13/ossutilarm64
```

2. 运行以下命令修改文件执行权限。

```
chmod 755 ossutilarm64
```

3. 运行以下命令，生成配置文件。

```
./ossutilarm64 config
```

4. 根据提示设置配置文件。设置方法与Linux系统生成配置文件的方法相同。

2.3. 常用命令

2.3.1. appendfromfile（追加上传）

appendfromfile命令用于在已上传的追加类型文件（Appendable Object）末尾直接追加内容。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于追加上传的更多信息，请参见[追加上传](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-idx选项。关于这三个选项的具体用法，请参见[通用选项](#)。

命令格式

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

参数说明如下：

选项	说明
localfilename	本地文件完整路径。
bucketname	目标Bucket名称。

选项	说明
objectname	目标Object名称。追加上传时可直接将本地文件名称保留为Object名称，也可以自定义上传至Bucket后的Object名称。
--meta	设置Object的meta信息。仅支持在首次追加上传时附加此选项，例如 <code>--meta "x-oss-object-acl:private"</code> 。 Object的meta信息配置完成后，您可以通过 set-meta（管理文件元信息） 命令修改Object的meta信息。

使用示例

将本地根目录下的文件exampleobject.txt首次追加上传至目标存储空间examplebucket，然后直接在exampleobject.txt文件末尾多次追加内容。

- 1. 上传文件exampleobject.txt，并指定文件读写权限为私有。

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

以下输出结果表明exampleobject.txt已追加上传至目标Bucket，此时文件大小为5 Byte。

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

- 2. 在exampleobject.txt文件末尾追加文件dest.txt的内容。

如果需要在exampleobject.txt文件末尾多次追加内容，请相应替换如下示例中的待追加上传文件dest.txt。

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

以下输出结果已在exampleobject.txt文件后追加了内容，此时文件大小为150 Byte。

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

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要向另一个阿里云账号下，华东2（上海）下名为examplebucket的存储空间追加上传exampleobject.txt文件，示例如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.2. bucket-cname（自定义域名）

您可以通过bucket-cname命令创建域名所有权验证所需的CnameToken以及管理Cname配置。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

创建CnameToken

 **说明** 仅ossutil1.7.13及以上版本支持创建CnameToken。

- 命令格式

```
./ossutil64 bucket-cname --method put --item token oss://bucketname domainname
```

参数说明如下：

参数	说明
bucketname	填写待绑定Cname配置的Bucket名称。
domainname	填写自定义域名。

- 使用示例

以下示例用于为examplebucket待绑定的自定义域名example.com创建CnameToken。

```
./ossutil64 bucket-cname --method put --item token oss://examplebucket example.com
```

输出结果如下：

```
<CnameToken>
  <Bucket>examplebucket</Bucket>
  <Cname>example.com</Cname>
  <Token>4db41c3ad0c4c4b690d4c17fb34e****</Token>
  <ExpireTime>Thu, 26 May 2022 19:14:12 GMT</ExpireTime>
</CnameToken>
0.270654(s) elapsed
```

CnameToken创建完成后，您需要添加TXT记录。具体步骤，请参见[验证域名所有权](#)。

查看CnameToken

- 命令格式

```
./ossutil64 bucket-cname --method get --item token oss://bucketname domainname
```

- 使用示例

以下示例用于查看CnameToken。

```
./ossutil64 bucket-cname --method get --item token oss://examplebucket example.com
```

输出结果如下：

```
<CnameToken>
  <Bucket>examplebucket</Bucket>
  <Cname>example.com</Cname>
  <Token>fbf997e8a2d48cb2177ed25180ee****</Token>
  <ExpireTime>Thu, 26 May 2022 19:53:19 GMT</ExpireTime>
</CnameToken>
0.144356(s) elapsed
```

添加Cname配置

 **注意** 添加Cname配置之前，您需要先创建CnameToken。具体步骤，请参见[创建CnameToken](#)。

- 命令格式

```
./ossutil64 bucket-cname --method put oss://bucketname domainname
```

- 使用示例

以下示例用于为examplebucket添加Cname配置。

```
./ossutil64 bucket-cname --method put oss://examplebucket example.com
```

以下输出结果表明已成功添加Cname配置。

```
0.227834(s) elapsed
```

获取Cname配置

- 命令格式

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

- 使用示例

以下示例用于获取examplebucket的Cname配置，并将配置结果直接输出屏幕。

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

以下输出结果表明已成功获取Cname配置。Cname配置显示examplebucket绑定了两个自定义域名，分别为example.com和example.org，且域名处于已启用（Enabled）状态等信息。

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

删除CName配置

- 命令格式

```
./ossutil64 bucket-cname --method delete oss://bucketname domainname
```

- 使用示例

以下示例用于删除examplebucket的CName配置。

```
./ossutil64 bucket-cname --method delete oss://examplebucket example.com
```

以下输出结果表明已成功删除CName配置。

```
0.227389(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要为另一个阿里云账号下，华东1（杭州）地域下的testbucket添加CName配置，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.3. bucket-encryption（服务器端加密）

配置服务器端加密（即Bucket加密）后，OSS对上传的文件（Object）进行加密，再将得到的加密文件持久化保存。下载文件时，OSS自动将加密文件解密后返回给用户。本文介绍如何通过 bucket-encryption命令添加、修改、查询和删除Bucket的加密配置。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于服务器端加密的更多信息，请参见[服务器端加密](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

添加或修改Bucket加密配置

命令格式

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

参数说明如下：

参数	说明
bucketName	配置服务器端加密的目标Bucket。
--sse-algorithm	Bucket的加密方式。 取值： ◦ KMS：使用KMS托管密钥进行加解密，即SSE-KMS。 ◦ AES256：使用OSS完全托管密钥进行加解密，即SSE-OSS。 ❓ 说明 在OSS ON云盒使用场景中，仅支持AES256。
--kms-masterkey-id	当加密方式为SSE-KMS时，OSS使用默认托管的KMS CMK进行加密。如果您希望通过指定的KMS CMK进行加密，请通过此选项设置正确的CMK ID。 ❓ 说明 在OSS ON云盒使用场景中，不支持使用此选项。

使用示例

- 将examplebucket的服务器端加密方式设置为SSE-OSS，并指定加密算法为AES256。

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

- 将examplebucket的服务器端加密方式设置为SSE-KMS，指定CMK ID，并使用默认加密算法AES256进行加密。

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

- 以下输出结果表明examplebucket已成功配置服务器端加密。

```
0.856895(s) elapsed
```

获取Bucket加密配置

- 命令格式

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

- 使用示例

获取examplebucket的加密配置。命令如下：

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

以下输出结果表明examplebucket配置的服务器端加密方式为SSE-KMS，未指定CMK ID，且加密算法为AES256。

```
SSEAlgorithm:KMS
KMSMasterKeyID:
KMSEncryption:
```

删除Bucket加密配置

- 命令格式

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

- 使用示例

删除examplebucket的加密配置。命令如下：

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

以下输出结果表明已成功删除examplebucket的服务器端加密配置。

```
0.856686(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要为另一个阿里云账号下，华东1（杭州）名为examplebucket的存储空间配置AES256的加密方式，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.4. bucket-policy（授权策略）

Bucket Policy是基于资源的授权策略，Bucket拥有者可以通过Bucket Policy授权其他用户访问OSS指定资源。**bucket-policy**命令用于添加、修改、查询、删除Bucket授权策略（Bucket Policy）。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于Bucket Policy的更多信息，请参见[Bucket Policy概述](#)。

添加或修改Bucket Policy

添加或修改Bucket Policy前，需要在本地创建JSON格式的文件，并在JSON文件中配置Bucket Policy。单个JSON文件可以配置多条Bucket Policy，但所有Bucket Policy的总大小不能超过16 KB。

添加或修改Bucket Policy时，ossutil先从JSON格式的文件中读取Bucket Policy配置，然后将读取到Bucket Policy添加到指定的Bucket。添加Bucket Policy为覆盖语义，即新添加的Bucket Policy会覆写已有的Bucket Policy配置。

- 命令格式

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

参数说明如下：

参数	说明
bucketname	添加或修改Bucket Policy的目标存储空间名称。
local_json_file	配置Bucket Policy的本地JSON文件名称。

- 使用示例

- i. 在本地创建名为local_json_file文件，并根据使用场景写入不同的Bucket Policy。

Bucket Policy的常见配置示例如下：

 **说明** 以下为资源拥有者（即Resource中UID为 174649585760**** 的Bucket Owner）通过Bucket Policy授权指定用户（例如Principal中UID为 20214760404935**** 的RAM用户，匿名用户为 *）不同权限的示例。

- 仅允许指定IP的匿名用户访问目标存储空间examplebucket内的所有资源。

```
{
  "Statement": [
    {
      "Action": [
        "oss:GetObject",
        "oss:GetObjectAcl",
        "oss:ListObjects",
        "oss:RestoreObject",
        "oss:GetVodPlaylist",
        "oss:ListObjectVersions",
        "oss:GetObjectVersion",
        "oss:GetObjectVersionAcl",
        "oss:RestoreObjectVersion"
      ],
      "Resource": [
        "oss://examplebucket/*"
      ],
      "Principal": "*"
    }
  ],
  "Version": "1"
}
```

```
        "Condition": {
            "IpAddress": {
                "acs:SourceIp": [
                    "10.10.10.10"
                ]
            }
        },
        "Effect": "Allow",
        "Principal": [
            "*"
        ],
        "Resource": [
            "acs:oss*:174649585760****:examplebucket/*"
        ]
    },
    {
        "Action": [
            "oss:ListObjects",
            "oss:GetObject"
        ],
        "Condition": {
            "StringLike": {
                "oss:Prefix": [
                    "*"
                ]
            }
        },
        "IpAddress": {
            "acs:SourceIp": [
                "10.10.10.10"
            ]
        },
        "Effect": "Allow",
        "Principal": [
            "*"
        ],
        "Resource": [
            "acs:oss*:174649585760****:examplebucket"
        ]
    }
],
"Version": "1"
}
```

- 授权指定的RAM用户拥有读取examplebucket中 `hangzhou/2020` 和 `hangzhou/2015` 目录的只读权限。

```
{
  "Statement": [
    {
      "Action": [
        "oss:GetObject",
        "oss:GetObjectAcl",
        "oss:ListObjects",
        "oss:RestoreObject",
        "oss:GetVodPlaylist",
        "oss:ListObjectVersions",
        "oss:GetObjectVersion",
        "oss:GetObjectVersionAcl",
        "oss:RestoreObjectVersion"
      ],
      "Effect": "Allow",
      "Principal": [
        "20214760404935*****"
      ],
      "Resource": [
        "acs:oss*:174649585760****:examplebucket/hangzhou/2020/*",
        "acs:oss*:174649585760****:examplebucket/hangzhou/2015/*"
      ]
    },
    {
      "Action": [
        "oss:ListObjects",
        "oss:GetObject"
      ],
      "Condition": {
        "StringLike": {
          "oss:Prefix": [
            "hangzhou/2020/*",
            "hangzhou/2015/*"
          ]
        }
      },
      "Effect": "Allow",
      "Principal": [
        "20214760404935*****"
      ],
      "Resource": [
        "acs:oss*:174649585760****:examplebucket"
      ]
    }
  ],
  "Version": "1"
}
```

- 拒绝匿名用户访问examplebucket中 `hangzhou/2021/` 目录下的所有文件。


```

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

```

ii. 为examplebucket添加Bucket Policy。

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

以下输出结果表明已成功添加Bucket Policy。

```
1.125101(s) elapsed
```

获取Bucket Policy配置

● 命令格式

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

参数	说明
bucketname	获取Policy配置的目标Bucket名称。
local_json_file	用于存放Policy配置的本地JSON文件名称。如果未指定此参数，则Policy配置将直接输出到屏幕。

● 使用示例

获取examplebucket配置的Bucket Policy。

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

以下输出结果表明已成功获取Bucket Policy配置，并将其写入了本地JSON文件。

```
0.212407(s) elapsed
```

删除Bucket Policy配置

当您不再需要通过Bucket Policy授权其他用户访问您的OSS资源时，请删除已配置的Bucket Policy。

● 命令格式

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

● 使用示例

删除examplebucket中所有已配置的Bucket Policy。

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

以下输出结果表明examplebucket中已配置的所有Bucket Policy均已删除。

```
0.530750(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）名为examplebucket的存储空间配置Bucket Policy，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.5. bucket-tagging（存储空间标签）

您可以通过存储空间（Bucket）的标签功能，对Bucket进行分类管理，例如对拥有指定标签的Bucket设置访问权限等。**bucket-tagging**命令用于添加、修改、查询、删除Bucket的标签配置。

注意

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于Bucket标签的更多信息，请参见[存储空间标签](#)。

添加或修改Bucket标签

存储空间标签使用一组键值对（Key-Value）来标记Bucket，每个Bucket仅允许添加10组标签。只有Bucket的拥有者以及被授予Put Bucket Tags权限的用户才能为Bucket添加或修改标签，否则返回403 Forbidden错误，错误码为AccessDenied。

• 命令格式

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

参数说明如下：

参数	说明
bucketname	添加或修改标签信息的Bucket名称。
key#value	标签信息的Key-Value对。 ◦ Key和Value以井号（#）分隔，且Key和Value必须为UTF-8编码。 ◦ Key最大长度为64字符，不能以 <code>http://</code>、<code>https://</code>、<code>Aliyun</code> 为前缀，且不能为空。 ◦ Value最大长度为128字符，可以为空。

若Bucket未设置标签，此命令将为Bucket添加指定的标签；若Bucket已设置标签，此命令将覆盖Bucket已有标签。

• 使用示例

为名为examplebucket的Bucket配置两组标签，其中一组标签的Key为tag1，Value为test1；另一组标签的Key为tag2，Value为test2。

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

以下输出结果表明已成功添加Bucket标签信息。

```
0.300600(s) elapsed
```

获取Bucket标签

- 命令格式

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

- 使用示例

获取examplebucket的标签配置。

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

以下输出结果表明examplebucket配置了两组标签，其中一组标签的Key为tag1，Value为test1；另一组标签的Key为tag2，Value为test2。

index	tag key	tag value
0	"tag1"	"test1"
1	"tag2"	"test2"

0.283359(s) elapsed

删除Bucket标签

- 命令格式

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

- 使用示例

删除examplebucket所有包含的标签信息。

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

以下输出结果表明examplebucket中包含的所有标签信息均已删除。

0.530750(s) elapsed

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）名为examplebucket的存储空间配置标签信息，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.6. bucket-versioning（版本控制）

版本控制是针对存储空间（Bucket）级别的数据保护功能。开启版本控制后，针对数据的覆盖和删除操作将会以历史版本的形式保存下来。在对象（Object）被错误覆盖或者误删除后，您可以将Bucket中存储的Object恢复至任意时刻的历史版本。本文介绍通过bucket-versioning命令设置或获取版本控制状态。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于版本控制的功能详情，请参见[版本控制介绍](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

设置版本控制状态

- 命令格式

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

参数说明如下：

参数	说明
bucketname	待设置版本控制状态的目标Bucket名称。
versioning	为目标Bucket设置版本控制状态。取值如下： ◦ <i>enabled</i>: 开启版本控制。当Bucket处于开启版本控制状态时，OSS将为新上传的Object生成全局唯一的随机字符串版本ID。有关启用版本控制状态下Object相关操作的更多信息，请参见开启版本控制下Object的操作。 ◦ <i>suspended</i>: 暂停版本控制。当Bucket处于暂停版本控制状态时，OSS将为新上传的Object生成特殊字符串为“null”的版本ID。有关暂停版本控制状态下Object的相关操作的更多信息，请参见暂停版本控制下Object的操作。



注意 默认情况下，Bucket版本控制状态为“未开启”。一旦Bucket开启了版本控制，将无法返回至“未开启”状态。但是，您可以暂停Bucket版本控制。

- 使用示例

对目标存储空间examplebucket开启版本控制。

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

对目标存储空间examplebucket暂停版本控制。

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

以下输出结果表明examplebucket已成功设置版本控制状态。

```
0.261209(s) elapsed
```

获取版本控制状态

- 命令格式

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

- 使用示例

查询examplebucket的版本控制状态。

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

以下输出结果表明examplebucket已开启版本控制。

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

以下输出结果表明examplebucket已暂停版本控制。

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

以下输出结果表明examplebucket未开启版本控制。

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

相关操作

- 在开启版本控制的Bucket中上传文件的行为与未开启版本控制状态下的上传行为一致。但Bucket开启版本控制后，OSS会为新上传至Bucket的Object生成全局唯一的versionID。详情请参见[cp（上传文件）](#)。
- 开启Bucket的版本控制后，针对数据的覆盖和删除操作将会以历史版本的形式保存下来。您可以通过指定versionID的方式下载指定版本Object，详情请参见[cp（下载文件）](#)。您还可以通过指定versionID的方式将Object恢复至任意时刻的历史版本，详情请参见[cp（拷贝文件）](#)。

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）名为examplebucket的存储空间开启版本控制，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.7. cat（输出文件内容）

cat命令仅支持将存储空间（Bucket）内文件（Object）的内容输出到屏幕。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

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

 **注意** 仅建议使用此命令查看文本文件内容。

参数及选项说明如下：

配置项	说明
bucketname	Bucket名称。
objectname	Object名称。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <i>requester</i> 。
--version-id	Object的指定版本。仅适用于已开启或暂停版本控制状态Bucket下的Object。

使用示例

- 将未开启版本控制的目标存储空间examplebucket内名为test.txt的文件内容输出到屏幕

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

以下输出结果表明test.txt文件包含的信息，以及完成输出操作所用时长。

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

- 将已开启版本控制的目标存储空间examplebucket内名为exampleobject.txt文件的指定版本内容输出到屏幕

```
./ossutil64 cat oss://examplebucket/exampleobject.txt --version-id CAEQARiBgID8rumR2hYiI
GUyOTAYZGY2MzU5MjQ5Zj1hYzQzZjNlYTAyZDE3****
```

有关获取Object所有版本的具体操作，请参见[ls命令](#)。

以下输出结果表明指定版本的exampleobject.txt文件包含的信息，以及完成输出操作所用时长。

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

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要查看另一个阿里云账号下，华东2（上海）地域下目标存储空间examplebucket1下名为exampleobject1.txt文件内容，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.8. config（创建配置文件）

您可以通过config命令创建配置文件，用于存储OSS访问信息。您可以在使用其他命令时添加-c选项，ossutil将通过指定的配置来访问OSS。

 注意

本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

该命令有交互式和非交互式两种用法。交互式用法允许您在使用命令前，通过配置文件完成相关配置项。命令使用过程中，ossutil将直接读取配置文件中的配置。而非交互式用法则需要您在使用命令时，通过选项来指定需要的配置项。相比非交互式用法，交互式用法具有更高的安全性。

该命令的非交互式用法的命令格式如下：

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

各选项说明如下：

选项	说明
-e, --endpoint	填写Bucket所在地域的域名信息，详情请参见 访问域名和数据中心 。您也可以增加 http:// 或 https:// 指定ossutil访问OSS使用的协议，默认使用HTTP协议。
-i, --access-key-id	指定配置文件中Credentials选项下的AccessKey ID。有关如何查看AccessKey ID信息的具体步骤，请参见 获取AccessKey 。
-k, --access-key-secret	指定配置文件中Credentials选项下的AccessKey Secret。有关如何查看AccessKey Secret信息的具体步骤，请参见 获取AccessKey 。
-t, --sts-token	访问OSS使用的STSToken，仅在使用STS临时授权方式访问OSS时需要配置此项。有关如何生成STSToken的具体步骤，请参见 临时访问凭证 。
--ram-role-arn	RamRoleArn鉴权模式下的RAM角色ARN。仅在使用RAM用户的AssumeRole的方式访问时需要配置此项。
-L language	设置ossutil工具的语言。取值如下： - CH（默认值）：中文。设置为CH时，请确保您的系统编码方式为UTF-8。 - EN：英文。

选项	说明
<code>--output-dir</code>	指定输出文件所在的目录，输出文件目前包含cp命令批量拷贝文件出错时所产生的report文件。 默认值：当前目录下的 <code>ossutil_output</code> 目录。
<code>-c, --config-file</code>	ossutil工具的配置文件路径，ossutil启动时将从配置文件读取配置。

使用示例

● 交互式配置

```
./ossutil64 config
请输入配置文件名，文件名可以带路径（默认为：/home/user/.ossutilconfig，回车将使用默认路径。
如果用户设置为其它路径，在使用命令时需要将--config-file选项设置为该路径）：
未输入配置文件路径，将使用默认配置文件：/home/user/.ossutilconfig。
对于下述配置，回车将跳过相关配置项的设置，配置项的具体含义，请使用"help config"命令查看。
请输入endpoint：https://oss-cn-shenzhen.aliyuncs.com
请输入accessKeyId：yourAccessKeyId
请输入accessKeySecret：yourAccessKeySecret
请输入stsToken：yourStsToken
```

● 非交互式配置

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

如果您使用命令时输入了除 `-L language` 和 `-c, --config-file` 以外的任意选项，则进入非交互式模式，所有的配置项需使用选项来指定。

编辑配置文件

对于已经生成的配置文件，您也可以通过直接编辑配置文件来修改OSS访问信息。ossutil工具的配置文件格式如下：

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

- Bucket-Endpoint：为每个指定的Bucket单独配置Endpoint。
- Bucket-Cname：为每个指定的Bucket单独配置CNAME域名。关于CNAME域名的更多信息，请参见[绑定自定义域名](#)。
- AkService：当您希望使用ECS实例绑定的RAM角色操作OSS时，需配置此项。配置时仅需将EcsRamRoleTesting改为ECS实例绑定的角色名称即可。配置此项后，AccessKey ID、AccessKey Secret、STSToken可不配置。如果配置了AccessKey ID，则AkService的配置将不会生效，而是以配置的AccessKey ID、AccessKey Secret、STSToken进行身份校验。ECS实例绑定RAM角色请参见[授予实例RAM角色](#)。

注意

- 在新版本中，ossutil取消了交互式配置中关于Bucket-Endpoint和Bucket-Cname项的配置。您可以在配置文件中对每个Bucket单独指定Endpoint或CNAME。
- ossutil支持通过不同方式指定Endpoint，Endpoint生效优先级为--endpoint（命令选项）> Bucket-Cname > Bucket-Endpoint > endpoint（Credentials项）。

2.3.9. cors（跨域资源共享）

跨域资源共享CORS（Cross-Origin Resource Sharing）简称跨域访问，是HTML5提供的标准跨域解决方案，允许Web应用服务器进行跨域访问控制，确保跨域数据传输的安全性。本文介绍如何通过cors命令添加、修改、查询、删除存储空间（Bucket）的CORS配置。

说明

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于跨域资源共享的更多信息，请参见[跨域资源共享](#)。

添加或修改CORS配置

如果Bucket此前未添加过CORS配置，则使用该命令将为Bucket添加CORS配置。如果Bucket此前已添加CORS配置，则使用该命令将直接覆盖已有的CORS配置。

添加或修改CORS配置步骤如下：

1. 创建本地文件，并在文件中添加CORS规则。
2. ossutil先从本地文件中读取CORS配置，然后将读取到的CORS配置添加到指定的Bucket。

添加或修改CORS配置的命令格式及示例如下：

- 命令格式

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

参数说明如下：

参数	说明
bucketname	待添加或修改CORS配置的Bucket名称。
local_xml_file	配置CORS规则的本地文件名称，例如 <code>localfile.txt</code> 。

- 使用示例

- i. 在本地创建名为 `localfile.txt` 文件，并根据使用场景配置CORS规则。

如下示例表示在CORS规则中指定允许的跨域请求来源为 `www.aliyun.com`、允许的跨域请求方法为 `PUT` 以及浏览器对特定资源的预取（OPTIONS）请求返回结果的缓存时间为10000秒。

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

- ii. 为examplebucket添加CORS规则。

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

以下输出结果表明已成功添加CORS规则。

```
0.299514(s) elapsed
```

获取CORS配置

- 命令格式

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

参数说明如下：

参数	说明
bucketname	获取CORS配置的目标Bucket名称。

参数	说明
local_xml_file	用于存放CORS配置的本地文件名称，例如 <code>localfile.txt</code> 。如果未指定此参数，则CORS配置将直接输出到屏幕。

● 使用示例

获取目标存储空间examplebucket的CORS配置。

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

以下输出结果表明已成功获取CORS配置，并将其写入本地localfile.txt文件。

```
0.212407(s) elapsed
```

删除CORS配置

● 命令格式

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

● 使用示例

删除examplebucket的CORS配置。

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

以下输出结果表明已删除examplebucket的CORS配置。

```
0.530750(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）名为testbucket的Bucket添加CORS配置，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.10. cors-options（检测跨域请求）

跨域资源共享CORS（Cross-Origin Resource Sharing）简称跨域访问，是HTML5提供的标准跨域解决方案，允许Web应用服务器进行跨域访问控制，确保跨域数据传输的安全性。cors-options命令通过向OSS发送HTTP OPTIONS请求的方式，用来检测指定的跨域访问请求是否被允许。

说明

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于添加CORS配置的具体步骤，请参见[cors（跨域资源共享）](#)。

命令格式

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

参数和选项说明如下：

配置项	说明
bucketname	请求访问的Bucket名称。
objectname	请求访问的Object名称。
--acr-method	实际请求中用到的方法，对应请求头 Access-Control-Request-Method 。可选值为 GET、PUT、POST、DELETE和 HEAD。
--origin	标识跨域请求的来源域。例如 http://www.aliyun.com 。
--acr-headers	实际请求中用到的除了简单头部之外的Header，对应请求头 Access-Control-Request-Headers 。您可以在请求中设置多个Header，多个Header之间使用英文逗号（,）分隔。例如 --acr-headers "header1,header2,header3" 。

使用示例

以下示例用于检测目标存储空间examplebucket是否允许指定来源域为 http://www.aliyun.com ，请求方法为 PUT 的跨域访问请求。

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

如果examplebucket允许该跨域访问请求，则返回结果将包含如下信息。

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

如果examplebucket不允许该跨域访问请求，则返回结果将包含如下信息。

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

通用选项

例如，您需要将另一个阿里云账号下，华东2（上海）地域下源存储空间examplebucket的文件夹srcfolder同步至目标存储空间testbucket的文件夹examplefolder，命令如下：

```
./ossutil64 sync oss://examplebucket/srcfolder/ oss://testbucket/examplefolder/ -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

例如，您需要检测另一个阿里云账号下，华东1（杭州）地域名为testbucket的Bucket是否允许指定来源域为 `http://www.alibabacloud.com`，请求方法为 `PUT` 的跨域访问请求，命令如下：

```
./ossutil64 cors-options --acr-method put --origin "http://www.alibabacloud.com" oss://testbucket -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.11. cp（上传、下载和拷贝文件）

2.3.11.1. 简介

`cp`命令用于上传、下载、拷贝文件。

有关`cp`命令用于上传文件时的命令格式及使用示例的更多信息，请参见[上传文件](#)。

有关`cp`命令用于下载文件时的命令格式及使用示例的更多信息，请参见[下载文件](#)。

有关`cp`命令用于拷贝文件时的命令格式及使用示例的更多信息，请参见[拷贝文件](#)。

2.3.11.2. 上传文件

您可以通过`cp`命令将您的本地文件或文件夹上传至OSS。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的`./ossutil64`替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 使用`cp`命令上传文件时，默认使用分片上传和断点续传。如果您因意外中断了文件上传的过程，且未继续完成该文件的上传，则已上传的部分会以碎片（Part）的形式存储在OSS的存储空间（Bucket）中。如果您不再需要这些Part，建议您通过以下方式删除，以免产生额外的存储费用。
 - 手动删除Part，请参见[删除碎片](#)。
 - 通过生命周期规则自动删除Part，请参见[设置生命周期规则](#)。

- 当您在OSS ON云盒中使用该命令时：
 - a. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - b. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
file_url	本地文件路径。例如Linux系统文件路径 /localfolder/examplefile.txt ， Windows系统文件路径 D:\localfolder\examplefile.txt 。
cloud_url	OSS文件路径。格式为 oss://bucketname/objectname 。例如 oss://examplebucket/examplefile.txt 。
-r, --recursive	递归操作。当指定该选项时，ossutil会对Bucket下所有符合条件的Object进行操作，否则只对指定的单个Object进行操作。
-f --force	强制操作，不进行询问提示。
-u, --update	只有当目标文件不存在，或源文件的最后修改时间晚于目标文件时，ossutil才会执行上传操作。
--maxupspeed	最大上传速度，单位为KB/s。默认值为0，表示不限制上传速度。

配置项	说明
--enable-symlink-dir	上传链接子目录，默认不上传。
--disable-all-symlink	上传时忽略所有的软链接子文件以及软链接子目录。
--disable-ignore-error	批量操作时不忽略错误。
--only-current-dir	仅上传当前目录下的文件，忽略子目录及子目录下的文件。
--bigfile-threshold	设置断点续传文件的大小阈值，单位为字节。 默认值：100 MB 取值范围：0~9223372036854775807
--part-size	设置分片大小，单位为字节。默认情况下ossutil会根据文件大小自行计算合适的分片大小值。 取值范围：1~9223372036854775807
--checkpoint-dir	指定断点续传记录信息所在的目录。断点续传操作失败时，ossutil会自动创建名为 <code>.ossutil_checkpoint</code> 的目录，并在该目录下记录checkpoint信息，断点续传成功后会删除该目录。如果指定了该选项，请确保指定的目录可以被删除。
--encoding-type	文件名称的编码方式。取值为 <code>url</code> 。如果不指定该选项，则表示文件名称未经过编码。
--include	包含符合指定条件的所有文件。
--exclude	不包含任何符合指定条件的文件。
--meta	文件的元信息。包括部分HTTP标准属性（HTTP Header）以及以 <code>x-oss-meta-</code> 开头的用户自定义元数据（User Meta）。文件元信息格式为 <code>header:value#header:value</code> ，示例为 <code>Cache-Control:no-cache#Content-Encoding:gzip</code> 。有关OSS支持元信息的更多介绍，请参见 管理文件元信息 。
--acl	文件的读写权限ACL。取值如下： <code>default</code>：继承Bucket的读写权限。 <code>private</code>（默认值）：只有该Bucket的拥有者可以对该Bucket内的文件进行读写操作，其他人无法访问该Bucket内的文件。 <code>public-read</code>：只有Bucket拥有者可以对该Bucket内的文件进行写操作，其他用户（包括匿名访问者）都可以对该Bucket中的文件进行读操作。这有可能造成您数据的外泄以及费用激增，如果被人恶意写入违法信息还可能侵害您的合法权益。除特殊场景外，不建议您配置此权限。 <code>public-read-write</code>：任何人（包括匿名访问者）都可以对该Bucket内文件进行读写操作。这有可能造成您数据的外泄以及费用激增，请谨慎操作。
--snapshot-path	指定保存上传文件时的快照信息所在的目录。在下次上传文件时，ossutil会读取指定目录下的快照信息进行增量上传。

配置项	说明
--disable-crc64	关闭CRC64数据校验。默认情况下，ossutil进行数据传输时都会打开CRC64校验。
--disable-dir-object	表示上传文件时不为目录生成Object。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <code>requester</code> 。
--tagging	上传文件时设置标签信息，格式为 <code>TagkeyA=TagvalueA&TagkeyB=TagvalueB....</code> 。
-j, --jobs	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。
--parallel	单文件操作时的并发任务数，取值范围为1~10000。如果不设置此选项，默认由ossutil根据操作类型和文件大小自行决定。

从以上命令格式中的-j, --jobs和--parallel选项得知，当ossutil自行设置的默认并发数达不到用户的性能要求时，您可以自行调整这两个选项来升降性能。默认情况下，ossutil会根据文件大小来计算parallel个数。当批量上传大文件时，实际的并发数为jobs个数乘以parallel个数。

- 当ECS虚拟机或者服务器在网络、内存、CPU等资源不是特别大的情况下，建议将并发数调整到100以下。如果网络、内存、CPU等资源没有占满，可以适当增加并发数。
- 由于线程间资源切换及抢夺等原因，如果并发数过大，ossutil上传性能可能会下降。此外，并发数过大还可能产生EOF错误。所以请根据实际的机器情况调整-j, --jobs和--parallel选项的数值。如果要进行压测，可在一开始时调低这两项数值，然后逐渐调大直至找到最优值。

示例环境

本文以Linux系统为例，将本地文件或文件夹上传至OSS中。您在实际使用中，请根据您的系统和使用环境修改对应参数。本文涉及的通用示例说明如下：

- 本地文件：examplefile.txt（根目录下的文件）
- 本地文件夹：localfolder（根目录下的文件夹）
- 目标Bucket：examplebucket
- 目标Bucket指定目录：desfolder

简单上传

ossutil支持将本地文件上传到OSS，示例如下：

- 上传单个文件

上传文件时，如果不指定上传至OSS的文件名，则默认使用原文件名进行保存；如果指定文件名，则按照指定的文件名保存在OSS中。

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

- 仅上传文件夹内的文件

使用cp命令时增加-r选项，可以只将本地文件夹内的文件上传到OSS指定路径。

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

- 上传文件夹及文件夹内的文件

使用`cp`命令时增加`-r`选项，并在OSS指定路径下添加本地文件夹名称，可以将本地文件夹以及文件夹内的文件上传到OSS指定路径。

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

- 上传单个文件并指定`--meta`选项

上传文件的同时可以使用`--meta`选项设置文件的meta信息，其内容格式为 `header:value#header:value` ...。

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

- 上传文件夹并跳过已有文件

批量上传失败重传时，可以指定`--update`（可缩写为`-u`）选项跳过已经上传成功的文件，实现增量上传。示例如下：

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

- 上传文件到开通了请求者付费模式的Bucket

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

- 仅上传当前目录下的文件，忽略子目录

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

- 上传时不为目录生成Object

OSS的目录是用一个0 KB大小，名称以正斜线（/）结尾的Object模拟的。上传时增加`--disable-dir-object`参数，ossutil不会为该目录生成Object，但您仍可以在OSS控制台看到对应的目录结构。当您删除目录内的文件时，该目录也会消失。

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

- 上传软链接子目录下的文件

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

- 上传时忽略所有的软链接子文件以及软链接子目录

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

上传时限速

您可以在上传文件时，结合`--maxupspeed`选项来限制上传的最大速度，单位为KB/s。示例如下：

- 上传文件并设置限速为1 MB/s

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

- 上传文件夹并设置限速为1 MB/s

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

上传并设置对象标签

您可以在上传文件时，通过--tagging选项设置文件对象标签，多个标签以and（&）符号隔开。示例如下：

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

关于对象标签的更多说明请参见[object-tagging（对象标签）](#)。

上传并指定存储类型

您可以在上传文件时，通过--meta选项设置文件存储类型。存储类型可选值为：

- Standard：标准存储
- IA：低频访问
- Archive：归档存储

 **说明** 在OSS ON云盒使用场景中，仅支持Standard类型。

如果上传时未指定存储类型，则以存储空间的存储类型为准。更多信息，请参见[存储类型介绍](#)。

上传并指定存储类型的示例如下：

- 上传单个文件并指定存储类型为低频访问类型

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

- 上传文件夹并指定存储类型为标准存储类型

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

上传并指定读写权限ACL

您可以在上传文件时，通过--meta选项设置文件的ACL。文件ACL可选值为：

- default：继承Bucket（默认）
- private：私有
- public-read：公共读
- public-read-write：公共读写

示例如下：

- 上传单个文件并指定ACL为私有

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

- 上传文件夹并指定ACL为公共读

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

上传并指定加密方式

您可以在上传文件时指定文件的服务器端加密方式，将文件加密后保存在Bucket内。示例如下：

- 上传文件并指定加密方式为OSS完全托管，加密算法为AES256

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

- 上传文件并指定加密方式为KMS，不指定CMK ID

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

使用KMS加密时，会产生少量KMS密钥使用费用，详情请参见[KMS计费标准](#)。

- 上传文件并指定加密方式为KMS，指定CMK ID

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

有关服务器端加密功能介绍，请参见[服务器端加密](#)。

上传并生成快照

批量上传时，如果指定--snapshot-path选项，ossutil在指定的目录下生成文件上传的快照，记录成功上传的文件的本地lastModifiedTime，在下次上传时通过比较lastModifiedTime来决定是否跳过相同文件，所以在使用该选项时，请确保两次上传期间没有其他用户更改了OSS上对应的Object。--snapshot-path选项适用于加速增量上传的场景。示例如下：

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

注意

- ossutil不会主动删除snapshot-path文件夹中的快照信息。为避免快照信息过多，请定期删除snapshot-path文件夹内无用的快照。
- 由于读写snapshot信息需要额外开销，当待批量上传的文件数量较少、网络状况较好、或有其他用户操作相同Object时，不建议使用该选项。针对以上情况，您可以使用--update选项来增量上传。
- --update选项和--snapshot-path选项可以同时使用，ossutil会优先根据snapshot-path信息判断是否跳过此文件，如果不满足跳过条件，再根据--update判断是否跳过此文件。

批量上传符合条件的文件

批量上传时，如果指定--include和--exclude选项，ossutil会批量上传符合指定条件的文件。

--include和--exclude选项支持格式：

- 通配符星号 (*)：匹配所有字符。例如：*.txt表示匹配所有TXT格式的文件。
- 通配符问号 (?)：匹配单个字符。例如：abc?.jpg表示匹配所有文件名为abc+任意单个字符的JPG格式的文件，如abc1.jpg。
- [sequence]：匹配序列的任意字符，例如：abc[1-5].jpg表示匹配文件名为abc1.jpg~abc5.jpg的文件。
- [!sequence]：匹配序列外的任意字符，例如：abc[!0-7].jpg表示匹配文件名为abc0.jpg~abc7.jpg的文件。

一条规则中可以包含多个include（包含）和exclude（排除）条件，且每个文件从左到右逐一运用每个规则，直至最后才能最终确定匹配的结果。例如指定生效的文件夹中包含名为test.txt的文件，匹配不同的规则产生的结果如下。

- 规则一：`--include "*"test*" --exclude "*.txt"`，当规则匹配到`--include "*"test*"` ，匹配的结果为`test.txt`符合条件；当规则继续匹配到`--exclude "*.txt"`时，因`test.txt`文件名包含.txt，所以被排除。则匹配的最终结果为`test.txt`不符合条件。
- 规则二：`--exclude "*.txt" --include "*"test*"` ，当规则匹配到`--exclude "*.txt"`，匹配的结果为`test.txt`不符合条件；当规则继续匹配到`--include "*"test*"` ，因`test.txt`文件名包含test，所以符合条件。则匹配的最终结果为`test.txt`符合条件。
- 规则三：`--include "*"test*" --exclude "*.txt" --include "te?t.txt"` ，当规则匹配到`--include "*"test*"` ，匹配的结果为`test.txt`符合条件；当规则继续匹配到`--exclude "*.txt"`时，因`test.txt`文件名包含.txt，所以被排除；当规则最后匹配到`--include "te?t.txt"`时，因`test.txt`符合条件，所以被包含。则匹配的最终结果为`test.txt`符合条件。

 **注意** 设定条件时不支持目录格式，例如 `--include "/usr/test/.jpg"`。

示例如下：

- 上传所有文件格式为TXT的文件

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

- 上传所有文件名包含abc且不是JPG和TXT格式的文件

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

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要将本地文件exampleobject.txt上传至另一个阿里云账号下，华东2（上海）地域的存储空间examplebucket的destfolder目录下，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.11.3. 下载文件

您可以通过cp命令将OSS的文件（Object）下载到本地。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
cloud_url	OSS文件路径。格式为 oss://bucketname/objectname 。例如 oss://examplebucket/examplefile.txt 。
file_url	本地文件路径。例如Linux系统文件路径 /localfolder/examplefile.txt ， Windows系统文件路径 D:\localfolder\examplefile.txt 。
-r, --recursive	递归操作。当指定该选项时，ossutil会对Bucket下所有符合条件的Object进行操作，否则只对指定的单个Object进行操作。
-f --force	强制操作，不进行询问提示。
-u, --update	只有当目标文件不存在，或源文件的最后修改时间晚于目标文件时，ossutil才会执行下载操作。
--maxdownspeed	最大下载速度，单位为KB/s。默认值为0，表示不限制下载速度。
--disable-ignore-error	批量操作时不忽略错误。
--only-current-dir	仅下载当前目录下的文件，忽略子目录及子目录下的文件。

配置项	说明
--bigfile-threshold	设置断点续传文件的大小阈值，单位为字节。 默认值：100 MB 取值范围：0~9223372036854775807
--part-size	设置分片大小，单位为字节。默认情况下ossutil会根据文件大小自行计算合适的分片大小值。 取值范围：1~9223372036854775807
--checkpoint-dir	指定断点续传记录信息所在的目录。断点续传操作失败时，ossutil会自动创建名为 <code>.ossutil_checkpoint</code> 的目录，并在该目录下记录checkpoint信息，断点续传成功后会删除该目录。如果指定了该选项，请确保指定的目录可以被删除。
--range	下载目标文件的指定字段，并保存为一个新的文件，字段从0开始编号。 - 指定区间 例如指定为 <code>3-9</code>，表示下载文件的第3个字节到第9个字节（包含第3和第9字节）。 - 指定开始位置 例如指定为 <code>3-</code>，表示从第3个字节开始到文件结尾（包含第3个字节）。 - 指定结束位置 例如指定为 <code>-9</code>，表示从0字节到第9个字节（包含第9个字节）。
--encoding-type	文件名称的编码方式。取值为 <code>url</code> 。如果不指定该选项，则表示文件名称未经过编码。
--include	包含符合指定条件的所有文件。 ❓ 说明 不支持通过该选项指定为目录的形式，例如 <code>./ossutil64 cp oss://examplebucket/destfolder/ localfolder/ --include dir/</code>，否则报错 <code>--include or --exclude does not support format containing dir info</code>。
--exclude	不包含符合指定条件的文件。 ❓ 说明 不支持通过该选项指定为目录的形式，例如 <code>./ossutil64 cp oss://examplebucket/destfolder/ localfolder/ --exclude dir/</code>，否则报错 <code>--include or --exclude does not support format containing dir info</code>。
--meta	设置文件的元信息，格式为 <code>header:value#header:value</code> ，示例为 <code>Cache-Control:no-cache#Content-Encoding:gzip</code> 。有关元信息的介绍，请参见 set-meta（管理文件元信息） 。

配置项	说明
--acl	文件的读写权限ACL。取值如下： • <i>default</i>：继承Bucket的读写权限。 • <i>private</i>（默认值）：只有该Bucket的拥有者可以对该Bucket内的文件进行读写操作，其他人无法访问该Bucket内的文件。 • <i>public-read</i>：只有Bucket拥有者可以对该Bucket内的文件进行写操作，其他用户（包括匿名访问者）可以对该Bucket中的文件进行读操作。这有可能造成您数据的外泄以及费用激增，请谨慎操作。 • <i>public-read-write</i>：任何人（包括匿名访问者）都可以对该Bucket内文件进行读写操作。这有可能造成您数据的外泄以及费用激增，如果被人恶意写入违法信息还可能会侵害您的合法权益。除特殊场景外，不建议您配置公共读写权限。
--snapshot-path	指定保存下载文件时的快照信息所在的目录。在下次下载文件时，ossutil会读取指定目录下的快照信息进行增量下载。
--disable-crc64	关闭CRC64数据校验。默认情况下，ossutil进行数据传输时都会打开CRC64校验。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <code>requester</code> 。
--partition-download	用于分区下载。该选项取值格式为 <code>分区编号: 总分区数</code> 。例如 <code>1: 5</code> ，表示当前ossutil下载分区1，总共有5个分区。分区号从1开始编号，Object的分区规则由工具内部算法决定。通过该选项将待下载的Object分成多个分区，并由多个ossutil命令并行下载，各个ossutil命令下载各自的分区。
-j, --job	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。
--parallel	单文件操作时的并发任务数，取值范围为1~10000。如果不设置此选项，默认由ossutil根据操作类型和文件大小自行决定。
--version-id	下载指定版本的文件，仅支持在已开启版本控制的Bucket内使用。

从以上命令格式中的-j, --jobs和--parallel选项得知，当ossutil自行设置的默认并发数达不到用户的性能要求时，您可以自行调整这两个选项来升降性能。默认情况下，ossutil会根据文件大小来计算parallel个数。当批量下载大文件时，实际的并发数为jobs个数乘以parallel个数。

- 当ECS虚拟机或者服务器在网络、内存、CPU等资源不是特别大的情况下，建议将并发数调整到100以下。如果网络、内存、CPU等资源没有占满，可以适当增加并发数。
- 由于线程间资源切换及抢夺等原因，如果并发数过大，ossutil下载性能可能会下降。此外，并发数过大还可能产生EOF错误。所以请根据实际的机器情况调整-j, --jobs和--parallel选项的数值。如果要进行压测，可在一开始时调低这两项数值，然后逐渐调大直至找到最优值。

示例环境

本文以Linux系统为例，将OSS内的文件下载到本地。您在实际使用中，请根据您的系统和使用环境修改对应参数。示例环境如下：

- 操作系统：Linux
- 目标Bucket：examplebucket
- 目标Bucket指定目录：destfolder

- 本地文件：examplefile.txt（根目录下的文件）
- 本地文件夹：localfolder（根目录下的文件夹）

下载单个文件

下载文件时如果未指定文件名，则默认使用原文件名保存在本地。如果指定文件名，则按照指定文件名保存在本地。

- 沿用原文件名保存文件

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

- 按指定文件名保存文件

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

下载多个文件

ossutil不支持通过直接指定多个文件名的方式下载多个文件。您可以通过以下方式下载多个文件：

- 下载某一类文件

如果您希望同时下载多个文件名规律的文件，例如这些文件均包含相同前缀或者后缀，此时您可以结合 `--include` 和 `--exclude` 选项在下载时选择符合条件的文件。

- 下载所有文件格式不为 *JPG* 的文件

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

- 下载所有文件名包含 *abc* 且不是 *JPG* 和 *TXT* 格式的文件

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

- 下载文件夹（包含子目录）

下载某一个文件夹（包含子目录）下所有文件的示例如下：

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

当批量下载失败或者需要增量下载文件时，可以通过指定 `--update`（可缩写为 `-u`）选项选择跳过已经成功下载的文件。如果本地与OSS没有同名文件，或本地同名文件的最后修改时间早于OSS文件，ossutil会下载该文件。如果本地已有同名文件，且文件的最后修改时间晚于OSS内文件时，ossutil会跳过该文件。示例如下：

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

- 下载文件夹（不包含子目录）

如果您仅希望下载当前目录而忽略子目录下的文件时，可以通过指定 `--only-current-dir` 选项。示例如下：

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

下载时限速

您可以在下载文件时，结合 `--maxdownspeed` 选项来限制下载的最大速度，单位为KB/s。示例如下：

- 下载文件并设置限速为1 MB/s

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

- 下载文件夹并设置限速为1 MB/s

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

范围下载

下载文件时，可以通过--range选项指定下载范围。例如将`examplefile.txt`的第10到第20个字符作为一个文件下载到本地，命令如下：

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

下载并生成快照信息

批量下载时，如果指定--snapshot-path选项，ossutil在指定的目录下生成文件下载的快照信息，在下次指定该选项下载时，ossutil会读取指定路径下的快照信息进行增量下载。详情请参见[上传并生成快照](#)。

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

在已开启版本控制的Bucket内下载指定版本文件

开启Bucket的版本控制后，针对数据的覆盖和删除操作将会以历史版本的形式保存下来。您可以在cp命令中增加--version-id选项下载指定版本文件，命令如下：

```
./ossutil64 cp oss://my-bucket/test.jpg localfolder/ --version-id CAEQARiBgID8rumR2hYiIGUy  
OTAYZGY2MzU5MjQ5Zj1hYzQzZjNlYTAyZDE3MDRk
```

注意

- 使用--version-id选项前，需使用ls --all-versions命令获取文件的versionid。
- --version-id选项仅支持在已开启版本控制的Bucket内使用。开启Bucket版本控制命令请参见[bucket-versioning（版本控制）](#)。

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要将另一个阿里云账号下，华东2（上海）地域下存储空间examplebucket根目录下的文件exampleobject.txt下载至本地目录localfolder下，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.11.4. 拷贝文件

您可以通过cp命令将您OSS内的文件（Object）拷贝至相同存储空间（Bucket）的其他目录，或者拷贝至同一地域（Region）下的其他Bucket。目前只支持拷贝文件，不支持拷贝未完成的分片（Part）。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 不支持通过cp命令跨账号或者跨地域拷贝文件。如果您需要跨账号或者跨地域拷贝（迁移）文件，请使用[ossimport](#)或者[在线迁移服务](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
cloud_url	分别填写OSS源文件和目标文件路径。格式为 oss://bucketname/objectname 。例如，将目标存储空间 examplebucket 下的源文件srcobject.jpg拷贝到相同存储空间下的目标文件destobject.jpg，则源文件路径填写为 oss://examplebucket/srcobject.jpg ，目标文件路径填写为 oss://examplebucket/destobject.jpg 。
-r, --recursive	递归操作。当指定该选项时，ossutil会对Bucket下所有符合条件的Object进行操作，否则只对指定的单个Object进行操作。
-f --force	强制操作，不进行询问提示。
-u, --update	只有当目标文件不存在，或源文件的最后修改时间晚于目标文件时，ossutil才会执行拷贝操作。

配置项	说明
--disable-ignore-error	批量操作时不忽略错误。
--only-current-dir	仅拷贝当前目录下的文件，忽略子目录及子目录下的文件。
-bigfile-threshold	设置断点续传文件的大小阈值，单位为字节。 默认值：100 MB 取值范围：0~9223372036854775807
--part-size	设置分片大小，单位为字节。默认情况下ossutil会根据文件大小自行计算合适的分片大小值。 取值范围：1~9223372036854775807
--checkpoint-dir	指定断点续传记录信息所在的目录。断点续传操作失败时，ossutil会自动创建名为 <code>.ossutil_checkpoint</code> 的目录，并在该目录下记录checkpoint信息，断点续传成功后会删除该目录。如果指定了该选项，请确保指定的目录可以被删除。
--encoding-type	文件名称的编码方式。取值为 <code>url</code> 。如果不指定该选项，则表示文件名称未经过编码。
--include	包含符合指定条件的所有文件。
--exclude	不包含任何符合指定条件的文件。
--meta	设置文件的元信息，格式为 <code>header:value#header:value</code> ，示例为 <code>Cache-Control:no-cache#Content-Encoding:gzip</code> 。关于元信息的介绍，请参见 set-meta（管理文件元信息） 。
--acl	文件的读写权限ACL。取值如下： <code>default</code>：继承Bucket的读写权限。 <code>private</code>（默认值）：只有该Bucket的拥有者可以对该Bucket内的文件进行读写操作，其他人无法访问该Bucket内的文件。 <code>public-read</code>：只有Bucket拥有者可以对该Bucket内的文件进行写操作，其他用户（包括匿名访问者）都可以对该Bucket中的文件进行读操作。这有可能造成您数据的外泄以及费用激增，如果被人恶意写入违法信息还可能侵害您的合法权益。除特殊场景外，不建议您配置此权限。 <code>public-read-write</code>：任何人（包括匿名访问者）都可以对该Bucket内文件进行读写操作。这有可能造成您数据的外泄以及费用激增，请谨慎操作。
--disable-crc64	关闭CRC64数据校验。默认情况下，ossutil进行数据传输时都会打开CRC64校验。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <code>requester</code> 。
-j, --jobs	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。

配置项	说明
--parallel	单文件操作时的并发任务数，取值范围为1~10000。如果不设置此选项，默认由ossutil根据操作类型和文件大小自行决定。
--version-id	拷贝指定版本的文件，仅支持在已开启版本控制的Bucket内使用。

从以上命令格式中的-j, --jobs和--parallel选项得知，当ossutil自行设置的默认并发数达不到用户的性能要求时，您可以自行调整这两个选项来升降性能。默认情况下，ossutil会根据文件大小来计算parallel个数。当批量拷贝大文件时，实际的并发数为jobs个数乘以parallel个数。

- 当ECS虚拟机或者服务器在网络、内存、CPU等资源不是特别大的情况下，建议将并发数调整到100以下。如果网络、内存、CPU等资源没有占满，可以适当增加并发数。
- 由于线程间资源切换及抢夺等原因，如果并发数过大，ossutil拷贝性能可能会下降。此外，并发数过大还可能产生EOF错误。所以请根据实际的机器情况调整-j, --jobs和--parallel选项的数值。如果要进行压测，可在一开始时调低这两项数值，然后逐渐调大直至找到最优值。

示例环境

本文以Linux系统为例，演示如何在不同目录或不同Bucket之间拷贝文件。您在实际使用中，请根据您的系统和使用环境修改对应参数。示例环境如下：

- 操作系统：Linux
- 源Bucket：examplebucket1
- 源Bucket指定目录1：srcfolder1
- 源Bucket指定目录2：srcfolder2
- 目标Bucket：examplebucket2
- 目标Bucket指定目录：desfolder

注意事项

拷贝文件时，如果源路径末尾没有携带正斜线 (/)，则拷贝与指定前缀匹配的所有文件到目标Bucket。如果源路径末尾携带了正斜线 (/)，则仅拷贝指定目录下的文件拷到目标Bucket。

假设源存储空间examplebucket1指定目录srcfolder1下包含以下文件：

```
srcfolder1/exampleobject1.txt
srcfolder1/exampleobject2.png
srcfolder1/dir1/
srcfolder1/dir1/exampleobject3.jpg
srcfolder1/dir2/
srcfolder1/dir2/exampleobject4.jpg
```

- 拷贝时指定的源路径不以正斜线 (/) 结尾

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

拷贝完成后，目标存储空间examplebucket2下将新增以下文件：

```
srcfolder1/exampleobject1.txt
srcfolder1/exampleobject2.png
srcfolder1/dir1/
srcfolder1/dir1/exampleobject3.jpg
srcfolder1/dir2/
srcfolder1/dir2/exampleobject4.jpg
```

- 拷贝时指定的源路径以正斜线 (/) 结尾

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

拷贝完成后，目标存储空间examplebucket2下将新增以下文件：

```
exampleobject1.txt
exampleobject2.png
dir1/
dir1/exampleobject3.jpg
dir2/
dir2/exampleobject4.jpg
```

简单拷贝文件

- 拷贝单个文件

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

- 拷贝增量文件

批量拷贝时，如果指定--update选项，只有当目标文件不存在，或源文件的最后修改时间晚于目标文件时，ossutil才会执行拷贝操作。命令如下：

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

该选项可用于当批量拷贝失败重传时，跳过已经拷贝成功的文件，实现增量拷贝。

- 重命名文件

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

通过cp命令重命名文件时，原文件仍存在，您可以在重命名后删除原文件。

- 仅拷贝当前目录下的文件，忽略子目录

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

- 修改文件meta信息

拷贝文件的同时可以使用--meta选项修改文件的meta信息，其内容格式为 `header:value#header:value...`。

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

- 从开通了请求者付费模式的Bucket拷贝文件到普通Bucket

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

拷贝符合指定条件的文件

您可以使用--include和--exclude参数，在拷贝时选定符合条件的文件。

- 拷贝所有文件格式不为JPG的文件

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

- 拷贝所有文件名包含abc且不是JPG和TXT格式的文件

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

修改文件存储类型

您可以在覆写文件时，添加--meta选项来修改文件存储类型。存储类型可选值为：

- Standard：标准存储
- IA：低频访问
- Archive：归档存储
- ColdArchive：冷归档存储

 说明 在OSS ON云盒使用场景中，仅支持Standard类型。

关于存储类型的更多信息，请参见[存储类型介绍](#)。

- 将指定文件的存储类型修改归档存储

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

- 将指定文件夹下所有文件的存储类型修改为标准存储

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

注意

- 不支持通过cp命令将存储类型为归档或冷归档的Object转换为标准、低频访问类型，必须先通过[restore（解冻文件）](#)命令将该文件解冻，之后再使用cp命令转换存储类型。
- 如果将Object转换为低频访问、归档存储或冷归档存储类型，且存储未满足指定周期，会产生存储不足规定时长容量费用。更多信息，请参见[存储费用](#)。
- 通过cp命令对超过100 MB的文件进行存储类型转换操作时，ossutil默认会根据文件大小自行计算合适的分片大小值。如果自动切片无法满足需求，您可以通过--part-size选项指定分片大小，但需确保分片数不能超过10000。

修改文件对象标签

您可以在覆写文件时，添加--tagging选项修改文件对象标签，多个标签以and（&）符号隔开。命令如下：

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

关于对象标签的更多信息，请参见[object-tagging（对象标签）](#)。

拷贝并加密文件

您可以在拷贝文件时指定文件的服务器端加密方式，将文件加密后保存在Bucket内，关于服务器端加密功能介绍的更多信息，请参见[服务器端加密](#)。

- 拷贝文件并指定加密方式为AES256

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

- 拷贝文件并指定加密方式为KMS

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket2/destfolder/ --meta=x-oss-server-side-encryption:KMS
```

 **注意** 使用KMS加密时，OSS会在KMS平台上为这个文件创建一个主密钥，会产生少量KMS密钥API调用费用。更多信息，请参见[KMS计费标准](#)。

- 拷贝文件并指定加密方式为KMS，指定CMK ID

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

在已开启版本控制的Bucket内恢复文件

开启Bucket的版本控制后，针对数据的覆盖和删除操作将会以历史版本的形式保存下来。您可以在cp命令中增加--version-id选项，将历史版本覆写为最新版本。示例如下：

```
./ossutil64 cp oss://examplebucket1/examplefile.txt oss://examplebucket2/ --version-id CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5Zj1hYzQzZjNlYTAyZDE3MDRk
```

 **注意**

- 使用--version-id选项前，需使用ls --all-versions命令获取文件的versionid。
- --version-id选项仅支持在已开启版本控制的Bucket内使用。关于开启Bucket版本控制的具体操作，请参见[bucket-versioning（版本控制）](#)。

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要将另一个阿里云账号下，华东2（上海）地域下源存储空间examplebucket根目录下的文件srcobject.png拷贝至目标存储空间destbucket，命令如下：


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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.12. create-symlink（创建软链接）

软链接功能用于快速访问存储空间（Bucket）内的常用文件（Object）。通过create-symlink创建软链接后，您可以通过软链接文件快速打开源文件，类似于Windows的快捷方式。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
cloud_url	软链接文件所在Bucket的完整路径。
target_object	软链接文件指向的目标Object所在Bucket的完整路径。软链接文件与目标Object必须属于同一个Bucket。
--encoding-type	对 cloud_url 以及 target_object 中包含的文件名称进行编码，取值为url。如果不指定该选项，则表示文件名称未经过编码。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为requester。

使用示例

使用此命令创建软链接时不会检查目标文件是否存在。如果目标文件存在，通过创建的软链接文件可直接访问目标文件。如果目标文件不存在，通过创建的软链接文件无法找到目标文件。当您无法判断目标文件是否存在时，请通过ls（[列举](#)）命令获取目标Bucket内包含的所有文件。

为已存在的目标文件创建软链接文件示例如下：

 **注意** 如果新添加的软链接文件名称与已有的软链接文件重名，则新添加的软链接文件将覆盖已有的软链接文件。

- 为目标存储空间examplebucket根目录下的exampleobject.jpg文件创建名为test.jpg的软链接文件，并将软链接文件保存至该Bucket的根目录。

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

- 在开启请求者付费模式下，为目标存储空间examplebucket根目录下的test.jpg文件创建名为example.jpg的软链接文件，并将软链接文件保存至该Bucket下的destfolder目录。

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

- 以下输出结果表明已为目标文件成功创建软链接文件。

```
0.106744(s) elapsed
```

软链接文件创建成功后，您可以通过[read-symlink（读取软链接）](#)或[stat（查看Bucket和Object信息）](#)命令获取软链接文件相关信息，例如ETag值、文件最后更新时间等。

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要为另一个阿里云账号下，华东2（上海）地域下目标存储空间testbucket下的exampleobject.png文件创建名为testobject.png的软链接，命令如下：

```
./ossutil64 create-symlink oss://testbucket/testobject.png oss://testbucket/exampleobject.png -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.13. du（获取大小）

du命令用于获取指定存储空间（Bucket）、文件目录下包含的所有Object的大小。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
bucketname	目标Bucket名称。
prefix	Bucket下的某个文件目录或指定前缀。

配置项	说明
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为requester。
--all-versions	获取所有版本Object的大小。不添加此选项时，默认查询当前版本Object的大小。
--block-size	定义输出结果中指定Bucket或目录下包含的Object大小，取值为KB、MB、GB或TB。不添加此选项时，默认以Byte为单位统计Object的大小。  注意 此选项仅在ossutil1.7.3及以上版本可用。

查询指定Bucket下所有版本Object的大小

以下命令用于查询examplebucket内所有版本Object的大小：

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

以下输出结果表明examplebucket内共有13个Object，其中12个Object的存储类型为Standard（标准存储），1个Object为Archive（归档存储），Object总大小为132116024 字节。

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

查询指定目录下所有当前版本Object的大小

以下命令用于查询examplebucket内指定目录dir下的当前版本Object大小，Object大小以GB为单位进行统计：

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

以下输出结果表明存储空间examplebucket内指定目录dir下共有5个Object，其存储类型均为Standard，文件总大小为0.0002 GB。

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

查询与前缀匹配的所有版本Object的大小

以下命令用于查询目标存储空间examplebucket下与前缀test匹配的所有版本Object的大小，Object大小以KB为单位进行统计：

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

以下输出结果表明examplebucket下与前缀test匹配的Object共有4个，其存储类型均为Standard，大小为448.1455 KB。

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

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要获取另一个阿里云账号下，华东2（上海）地域下名为testbucket的存储空间下所有版本Object的大小，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.14. getallpartsize（获取碎片大小）

进行分片上传操作时，执行getallpartsize命令可获取Bucket内所有未完成的Multipart任务产生的碎片（Part）信息。

② 说明

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于分片上传的更多信息，请参见[分片上传](#)。

命令格式

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

bucketname填写目标Bucket名称，该Bucket内存在因未完成的Multipart任务产生的碎片。

使用示例

以下示例用于列举目标存储空间examplebucket内所有未完成的Multipart任务产生的碎片。

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

以下返回结果表明已成功获取examplebucket内的碎片数量、碎片总大小以及碎片对应的UploadId等信息。

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

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要获取另一个阿里云账号下，华东1（杭州）名为testbucket的Bucket内所有未完成的Multipart任务产生的碎片，命令如下：

```
./ossutil64 getallpartsize oss://testbucket -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.15. hash（计算CRC64或MD5）

hash命令用于计算本地文件的MD5或CRC64。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
localfile	本地文件的完整路径。
--type	数据计算类型。取值如下： - md5：计算本地文件的MD5。 取值为MD5时，会同时输出文件的MD5以及Content-MD5值。Content-MD5值是在计算MD5值获得128比特位数字的基础上，对该数字进行Base64编码得到的值。关于Content-MD5的更多信息，请参见RFC1864。 - crc64（默认值）：计算本地文件的CRC64。 关于CRC64的计算标准，请参见ECMA-182标准。

上传文件到目标Bucket后，文件的CRC64和MD5值有如下注意事项：

- 您可以通过`stat`命令中 `X-Oss-Hash-Crc64ecma` 字段和 `Content-Md5` 字段分别获取该文件的CRC64和Content-MD5值。详情请参见[stat（查看Bucket和Object信息）](#)。
- 在OSS支持CRC64校验之前上传至Bucket内的文件，不支持通过`stat`命令查看文件的CRC64值。
- 对于通过追加上传以及分片上传方式上传的文件，不支持通过`stat`命令查看这两种类型文件的Content-MD5值。

使用示例

- 计算本地文件 `test.txt` 的CRC64

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

- 计算本地文件 `test.txt` 的MD5

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

2.3.16. help（获取帮助信息）

`help`命令用来获取命令的帮助信息，当您不清楚某个命令的用法时，建议您使用`help`命令获取该命令的帮助信息。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的`./ossutil64`替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

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

使用示例

- 获取所有命令帮助信息

```
./ossutil64 help
```

- 获取cp命令帮助信息

```
./ossutil64 help cp
```

- 查看所有选项信息

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

2.3.17. inventory（清单）

您可以使用清单功能获取存储空间（Bucket）中指定文件（Object）的数量、大小、存储类型、加密状态等信息。相对于GetBucket (ListObjects)接口，在海量Object的列举场景中，建议您优先使用清单功能。本文介绍如何通过inventory命令添加、查询、列举或删除Bucket清单规则。

说明

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于Bucket清单功能的更多信息，参见[存储空间清单](#)。

添加清单规则

添加清单规则步骤如下：

1. 生成RAM角色，该角色需拥有读取源Bucket所有文件和向目标Bucket写入文件的权限。配置角色的步骤请参见[创建可信实体为阿里云服务的RAM角色](#)。
2. 创建本地文件，并在文件中配置XML格式的清单规则。
3. ossutil先从本地文件中读取清单配置，然后将读取到清单配置添加到指定的Bucket。

- 命令格式

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

参数说明如下：

参数	说明
bucketname	添加清单规则的存储空间名称。
local_xml_file	配置清单规则的本地文件名称，例如 localfile.txt。

- 使用示例

- i. 创建本地文件localfile.txt，并根据使用场景写入不同的清单规则。

例如，将清单规则名称设置为inventorytest，以周为单位将清单报告导出至目标存储空间destbucket，清单内容中包含destbucket中所有文件的存储类型、最后更新时间以及分片上传状态等，并使用AES256加密算法加密清单文件。

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

 **注意** 您可以为Bucket添加多条清单规则，规则名称（ID）是清单配置的唯一标识。若添加的规则ID已存在，则返回409错误。

ii. 为examplebucket添加清单规则。

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

以下输出结果表明已成功添加清单规则。

```
0.299514(s) elapsed
```

查询指定清单规则

● 命令格式

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

参数说明如下：

参数	说明
bucketname	查询清单配置的目标Bucket名称。

参数	说明
inventory_id	清单规则名称。
local_xml_file	用于存放清单配置的本地文件名称，例如 <code>localfile.txt</code> 。如果未指定此参数，则清单配置将直接输出到屏幕。

● 使用示例

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

以下输出结果表明已成功查询examplebucket中配置规则ID为inventorytest的清单内容，并将清单结果写入本地localfile.txt文件。

```
0.212407(s) elapsed
```

查询所有清单规则

● 命令格式

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

参数说明如下：

参数	说明
bucketname	获取清单配置的目标Bucket名称。
local_xml_file	用于存放清单配置的本地XML文件名称。如果未指定此参数，则清单配置将直接输出到屏幕。
marker	清单过滤条件，只对与指定前缀匹配的Object生成清单文件。如果此项置空，表示对目标Bucket中的所有Object生成清单文件。

● 使用示例

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

以下输出结果表明已成功查询examplebucket下与dest前缀匹配的文件包含的所有清单规则，并将清单结果写入本地localfile.txt文件。

```
0.216897(s) elapsed
```

删除指定清单规则

● 命令格式

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

参数说明如下：

参数	说明
bucketname	删除清单配置的目标Bucket名称。
inventory_id	清单规则名称。

● 使用示例

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

以下输出结果表明已成功删除examplebucket配置规则ID为inventorytest的清单内容。

```
0.212407(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）名为examplebucket的存储空间配置清单规则，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.18. lcb（列举云盒信息）

lcb命令用于列举当前阿里云账号下关联的云盒信息。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

```
./ossutil64 lcb [--sign-version <value>][--region <value>][--cloudbox-id <value>]
```

各参数说明如下：

参数	说明
--sign-version	签名版本，取值为v4。
--region	云盒支持的地域。
--cloudbox-id	云盒ID。

使用示例

以下示例用于列举当前阿里云账号下关联的云盒信息。

```
./ossutil64 lcb --sign-version v4 --region cn-hangzhou --cloudbox-id cb-f8z7yvzgwfk19q0h***  
*
```

输出结果如下。

```
No          :0  
Id          :cb-f8z7yvzgwfk19q0h****  
Name       :oss-cn-heyuan-cloudbox-a  
Owner      :1*****4  
Region     :oss-cn-heyuan  
ControlEndpoint:cb-f8z7yvzgwfk19q0h****.cn-heyuan.oss-cloudbox-control.aliyuncs.com  
DataEndpoint  :cb-f8z7yvzgwfk19q0h****.cn-heyuan.oss-cloudbox.aliyuncs.com  
-----  
No          :1  
Id          :cb-wz9conqz5shwj945****  
Name       :oss-cn-shenzhen-cloudbox-b  
Owner      :1*****4  
Region     :oss-cn-shenzhen  
ControlEndpoint:cb-wz9conqz5shwj945****.cn-shenzhen.oss-cloudbox-control.aliyuncs.com  
DataEndpoint  :cb-wz9conqz5shwj945****.cn-shenzhen.oss-cloudbox.aliyuncs.com  
-----  
0.495359(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个阿里云账号时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要获取另一个阿里云账号下的云盒信息，示例如下：

```
./ossutil64 lcb --sign-version v4 --region cn-heyuan --cloudbox-id cb-u8s7yvzgwfk19q0h****  
-i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.19. lifecycle（生命周期）

配置生命周期规则后，OSS会定期将对象（Object）转储为低频访问、归档存储或冷归档存储类型，或将过期的Object和碎片删除，从而节省存储费用。本文介绍如何通过lifecycle命令添加、修改、查询、删除生命周期规则配置。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于生命周期规则的更多信息，请参见[生命周期规则介绍](#)。
- 当您在OSS ON云盒中使用该命令时：

- i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加 `--sign-version`、`--region`以及 `--cloudbox-id`选项。关于这三个选项的具体用法，请参见[通用选项](#)。
- 仅支持通过生命周期规则删除云盒Bucket中的Object，不支持转换Object存储类型。

添加或修改生命周期规则

添加或修改生命周期规则步骤如下：

1. 创建本地文件，并在文件中配置XML格式的生命周期规则。
2. ossutil先从本地文件中读取生命周期配置，然后将读取到生命周期配置添加到指定的Bucket。

- 命令格式

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

参数说明如下：

参数	说明
bucketname	添加或修改生命周期规则的存储空间名称。
local_xml_file	配置生命周期规则的本地文件名称，例如 <code>localfile.txt</code> 。

- 使用示例

 **注意** 您可以为Bucket添加多条生命周期规则，规则名称（ID）是生命周期配置的唯一标识。若添加的规则ID已存在，则返回409错误。

- i. 在本地创建名为 `localfile.txt` 文件，并根据使用场景写入不同的生命周期规则。

常见的生命周期规则如下：

■ 示例1

指定生命周期规则应用于目标存储空间examplebucket内的所有Object（即Prefix为空），在距其最后修改时间超过365天后全部删除；此外还指定了Prefix为test/，指示与前缀test/匹配的Object距其最后修改时间超过30天后转换为Archive存储类型。

有关生命周期支持转换的存储类型的更多信息，请参见[通过生命周期规则自动转换Object的存储类型](#)。

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

■ 示例2

指定生命周期规则应用于目标存储空间examplebucket内的所有Object（即Prefix为空），在其最后修改时间早于2019年12月30日的Object过期。

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

■ 示例3

在版本控制状态下，指定目标存储空间examplebucket内的所有Object距其最后修改时间超过10天后转换为IA存储类型，Object成为非当前版本60天后转换为Archive存储类型，Object成为非当前版本90天后删除。

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

ii. 为examplebucket添加生命周期规则。

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

以下输出结果表明已成功添加生命周期规则。

```
0.299514(s) elapsed
```

获取生命周期规则配置

● 命令格式

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

参数说明如下：

参数	说明
bucketname	获取生命周期规则配置的目标Bucket名称。
local_xml_file	用于存放生命周期规则配置的本地文件名称，例如 localfile.txt。如果未指定此参数，则生命周期规则配置将直接输出到屏幕。

● 使用示例

获取examplebucket配置的生命周期规则。

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

以下输出结果表明已成功获取生命周期规则配置，并将其写入本地localfile.txt文件。

```
0.212407(s) elapsed
```

删除生命周期规则配置

- 命令格式

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

- 使用示例

删除examplebucket的生命周期规则配置。

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

以下输出结果表明已删除examplebucket的生命周期规则配置。

```
0.530750(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）名为examplebucket的Bucket配置生命周期规则，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.20. listpart（列举碎片）

listpart命令用于列举未完成分片上传的文件（Object）包含的碎片（Part）信息。

② 说明

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于分片上传的更多信息，请参见[分片上传](#)。

命令格式

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

参数说明如下：

参数	说明
bucketname	目标Bucket名称。

参数	说明
objectname	未完成分片上传操作的Object名称。
uploadid	初始化Multipart Upload事件时返回的OSS服务器创建的全局唯一的Upload ID，用于标识本次Multipart Upload事件。您可以通过Upload ID进行Multipart Upload的中止或查询操作等。

 **注意** 您可以结合ls命令中的 `-m` 选项获取未完成分片上传的Object名称、初始化分片上传的时间以及Upload ID信息等。详情请参见[ls命令](#)。

使用示例

以下示例用于列举目标存储空间examplebucket下未完成分片上传的exampleobject.txt文件的碎片信息：

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

以下返回结果表明已成功获取exampleobject.txt文件的碎片数量、ETag值、碎片大小等信息。

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

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要获取另一个阿里云账号下，华东1（杭州）名为testbucket的Bucket下的testobject.txt文件所有未完成的Multipart任务产生的碎片，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.21. logging（日志转存）

访问对象存储OSS的过程中会产生大量的访问日志，您可以通过日志转存功能将这些日志按照固定命名规则，以小时为单位生成日志文件写入您指定的Bucket。您可以通过logging命令开启、查询、关闭Bucket的日志转存配置。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

- 关于日志转存的更多信息，请参见[日志转存](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加 `--sign-version`、`--region`以及 `--cloudbox-id`选项。关于这三个选项的具体用法，请参见[通用选项](#)。

开启日志转存

如果Bucket此前未开启过日志转存，则使用该命令将为Bucket开启日志转存。如果Bucket此前已开启日志转存，则使用该命令将直接覆盖已有的日志转存配置。

开启或修改日志转存配置的命令格式及示例如下：

• 命令格式

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

参数及选项说明如下：

配置项	说明
bucketname	生成日志的源Bucket名称。
targetbucket	存储日志文件的目标Bucket名称。
prefix	日志文件存储的目录。如果指定此项，则日志文件将保存在目标Bucket的指定目录下。如果不指定此项，则日志文件将保存在目标Bucket的根目录下。

 **注意** 生成日志的源Bucket和存储日志的目标Bucket可以相同也可以不同，但是必须位于同一账号下的相同地域。

• 使用示例

将源存储空间srcbucket产生的日志文件保存至目标存储空间destbucket的根目录下。

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

将源存储空间srcbucket产生的日志文件保存至目标存储空间destbucket的根目录下的子目录destdir。

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

以下输出结果表明已为srcbucket开启日志转存。

```
0.098601(s) elapsed
```

查看日志转存配置

• 命令格式

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

参数说明如下：

参数	说明
bucketname	待获取日志转存配置的Bucket名称。
local_xml_file	存放日志转存配置的本地文件名称，例如 <code>localfile.txt</code> 。如果未指定此参数，则日志转存配置将直接输出到屏幕。

- 使用示例
 - 获取存储空间examplebucket的日志转存配置，并将配置结果写入本地文件 `localfile.txt` 。

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

以下输出结果表明已成功获取日志转存配置。

```
0.212407(s) elapsed
```

- 获取存储空间examplebucket的日志转存配置，并将配置结果直接输出屏幕。

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

以下输出结果表明examplebucket的日志文件已转存至destbucket的根目录下。

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

关闭日志转存

在您关闭日志转存功能前，OSS的日志文件会一直生成。请结合生命周期规则及时清理不再需要的日志文件，以减少您的存储费用。详情请参见[基于最后一次修改时间的生命周期规则介绍](#)。

- 命令格式

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

bucketname表示待关闭日志转存功能的Bucket名称。

- 使用示例

关闭存储空间examplebucket的日志转存功能。

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

以下输出结果表明examplebucket已关闭日志转存功能。

```
0.212409(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要将另一个阿里云账号下，华东1（杭州）地域下源存储空间testbucket产生的日志转存至目标存储空间destbucket，命令如下：

```
./ossutil64 logging --method put oss://testbucket oss://destbucket -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkh7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.22. lrb（列举地域级别下的Bucket）

您可以通过lrb命令获取单个或多个地域（Region）下存储空间（Bucket）的基本信息，包括Bucket的名称、创建时间、存储类型以及个数等。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
conf_file	如果您希望获取多个Region下的Bucket信息时，请输入一个本地配置文件，并在文件中换行填写多个Region对应的Endpoint。
-e	如果您仅希望获取单个Region下的Bucket信息，请通过此选项指定该Region对应的Endpoint。

使用示例

- 获取单个地域的Bucket信息

获取华东1（杭州）地域下的Bucket信息。

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

获取与ossutil配置文件中指定Endpoint关联Region下的Bucket信息。有关ossutil配置文件中Endpoint的信息，请参见[config命令](#)。

```
./ossutil64 lrb
```

- 获取多个地域的Bucket信息

- i. 创建本地文件，并在文件中配置Endpoint信息。

在本地创建名为 localfile.txt 文件，并根据使用场景填写待获取Bucket信息所在Region对应的Endpoint。localfile.txt 配置如下：

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

- ii. 获取华东1（杭州）、华南1（深圳）以及华东2（上海）地域下的Bucket信息。

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

• 返回结果

获取Bucket信息成功后，返回结果中将包含Bucket的名称、创建时间、存储类型以及个数以及获取所用时长。以获取多个Region下的Bucket信息为例，其返回结果如下：

CreationTime	Region	StorageClass	BucketName
2021-07-06 14:21:09 +0800 CST	oss-cn-hangzhou	Standard	oss://examplebucket1
2021-07-06 14:21:44 +0800 CST	oss-cn-hangzhou	Standard	oss://examplebucket2
2021-06-16 18:32:32 +0800 CST	oss-cn-shanghai	Standard	oss://examplebucket3
2021-06-30 16:04:41 +0800 CST	oss-cn-shanghai	Standard	oss://examplebucket4
2021-07-07 12:33:35 +0800 CST	oss-cn-shenzhen	Standard	oss://examplebucket5

Bucket Number is: 5
0.124193(s) elapsed

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要获取另一个阿里云账号下，华东2（上海）地域下的Bucket信息，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.23. ls（列举账号级别下的资源）

您可以通过ls命令列举当前账号下的存储空间（Bucket）、对象（Object）和碎片（Part）信息。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

列举Bucket

- 命令格式

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

参数说明如下：

选项	说明
-s	列举结果仅返回Bucket的名称。
--limited-num	设定返回结果的最大个数。您可以使用此项结合marker对返回结果进行分页展示。
--marker	列举名称字母序排在marker之后的Bucket。

- 使用示例

- 列举所有Bucket

```
./ossutil64 ls
```

或

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

以下输出结果表明已成功列举当前账号下所有Bucket，包括Bucket名称、创建时间、所在地域、存储类型、数量等信息。

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

- 以精简模式列举所有Bucket

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

以下输出结果表明已成功列举当前账号下所有Bucket，仅包含Bucket名称以及Bucket数量。

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

- 列举字母序排在指定marker为examplebucketA之后的Bucket

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

以下输出结果表明已成功列举examplebucketA之后的2个Bucket。

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

列举Object

- 命令格式

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

参数说明如下：

选项	说明
bucketname	目标Bucket名称。
prefix	目标Object前缀。当您列举目标Bucket中指定前缀的Object时添加此项。
-s	列举结果仅返回Object的名称。
-d	仅列举Object和子目录，忽略子目录下的Object。
--limited-num	设定返回结果的最大个数。您可以使用此项结合--marker，对返回结果进行分页展示。
--marker	列举名称字母排序在marker之后的Object。
--include	列举符合指定条件的Object。例如 *.jpg 表示列举所有JPG格式的文件。更多信息，请参见 批量上传符合条件的文件 。
--exclude	列举不符合指定条件的Object。例如 *.txt 表示列举所有非TXT格式的文件。
--version-id-marker	列举Version ID字母排序在marker之后的Object版本。仅当Bucket开启版本控制后可用。
--all-versions	列举Object的所有版本，仅当Bucket开启版本控制后可用。

- 使用示例

- 列举examplebucket内所有Object

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

以下输出结果表明已成功列举examplebucket内所有Object的信息，包括文件最后更新时间（LastModifiedTime）、以字节为单位统计的文件大小（Size）、ETag值以及文件名称（ObjectName）。

其中，ETag值用于标识一个Object的内容。对于通过PutObject请求创建的Object，ETag值是其内容的MD5值；对于通过其他方式创建的Object，ETag值是基于一定计算规则生成的唯一值，但不是其内容的MD5值。

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

- 列举examplebucket内前缀为example的Object

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

以下输出结果表明已成功列举examplebucket内所有前缀为example的Object。

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

- 列举examplebucket内所有后缀名为.mp4的文件

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

以下输出结果表明已成功列举examplebucket内所有后缀名为.mp4的文件。

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

- 仅列举examplebucket根目录下Object和子目录

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

以下输出结果表明已成功列举examplebucket根目录下Object和子目录。

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

- 列举examplebucket内所有Object的所有版本

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

以下输出结果表明已成功列举examplebucket内所有Object的所有版本。

LastModifiedTime	Size (B)	StorageClass	ETag	IS-LATEST	DELETE-
VERSIONID					
MARKER	ObjectName				
2020-12-01 15:06:37 +0800 CST	114	Standard	61DE142E5AFF9A6748707D4A77BFB		
CFB	CAEQARiBgICUsOuR2hYiIDI3NWVjNmEyYmMONTTrkZWNIkYzY2VjMDMwZjFlMDA3			true	
false	oss://examplebucket/example.txt				
2020-06-11 11:03:37 +0800 CST	363812	Standard	E7581E5D2EBC56ECCB6FB6050B4C6		
545	CAEQARiBgIDztvuR2hYiIDNhYjRkN2M5NTA5OTRlN2Q4YTYzODQwMzQ4NDYwZDdm			true	
false	oss://examplebucket/examplefolder/photo.jpg				
2021-01-26 13:27:08 +0800 CST	0				
CAEQLxiBgIDd7NH0uRciIDA3Yzg0MTZjOWNlYzQ4ODZhMzVkZWE0MmE2NzBlYTYx				true	true
oss://examplebucket/image.png					
2020-12-01 15:06:45 +0800 CST	57374182	Standard	BE97B7AD7A2C1277B11221E5C9537		
544	CAEQLBiBgMDZiprwthciIDY2NGMONTNmZDE3ODRmZmVhZGM4YTUwZGQyNGU3ZjQ3			true	
false	oss://examplebucket/video.mp4				
2016-06-11 10:53:46 +0800 CST	118076	Standard	FFDB300F053AAF06F4C4C58A4869C		
427	CAEQARiBgID8rumR2hYiIGUyOTAYzGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3MDRk			false	
false	oss://examplebucket/example.txt				
2016-06-11 11:02:05 +0800 CST	345374	Standard	078A9852BCF81DC4811E6EDCBFD12		
1BE	CAEQARiBgICNz_ir2hYiIGJjZTBjNDQxYWRhNTQ2ZTNiNmMzYzQ1YzZmMDA5ZjUw			false	
false	oss://examplebucket/examplefolder/photo.jpg				
Object Number is: 6					
0.692000 (s) elapsed					

- 列举examplebucket根目录下 *example.txt* 的所有版本

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

以下输出结果表明已成功列举 *example.txt* 的所有版本。

```
LastModifiedTime          Size(B)  StorageClass  ETag
VERSIONID                                     IS-LATEST  DELETE-
MARKER  ObjectName
2020-12-01 15:06:37 +0800 CST          114      Standard  61DE142E5AFF9A6748707D4A77BFB
CFB      CAEQARiBgICUsOuR2hYiIDI3NWVjNmEyYmM0NTRkZWNiMTkxY2VjMDMwZjFlMDA3      true
false      oss://examplebucket/example.txt
2016-06-11 10:53:46 +0800 CST          114      Standard  61DE142E5AFF9A6748707D4A77BFB
CFB      CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5ZjlhYzQzZjNlYTAYZDE3MDRk      false
false      oss://examplebucket/example.txt
Object Number is: 2
0.361000(s) elapsed
```

列举Part

- 命令格式

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

参数说明如下：

选项	说明
bucketname	目标Bucket名称。 当您需要列举指定Bucket中的Object时添加此项。
prefix	列举指定前缀下的Part。
-s	列举结果仅返回UploadID和Object名称。
-d	列举Object和子目录，忽略子目录下的Object。
-m	列举Part。
-a	列举Object和Part。
--limited-num	设定返回结果的最大个数。您可以使用此项结合--upload-id-marker对返回结果进行分页展示。
--upload-id-marker	列举Upload ID字母排序在marker之后的Part。

- 使用示例

- 列举examplebucket内所有Part

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

以下输出结果表明已成功列举examplebucket内所有Part。

InitiatedTime	UploadID	ObjectName
2017-01-13 03:45:26 +0000 CST	15754AF7980C4DFB8193F190837520BB	oss://examplebucket/test.mp4
2017-01-13 03:45:13 +0000 CST	2A1F9B4A95E341BD9285CC42BB950EE0	oss://examplebucket/test.mp4
2017-01-13 03:45:01 +0000 CST	3998971ACAF94AD9AC48EAC1988BE863	oss://examplebucket/test.mp4
2017-01-20 11:16:21 +0800 CST	A20157A7B2FEC4670626DAE0F4C0073C	oss://examplebucket/object.exe

UploadId Number is:4
0.191289(s) elapsed

- 列举examplebucket内所有Object和Part

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

以下输出结果表明已成功列举examplebucket内所有Object和Part。

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

Object Number is:3

InitiatedTime	UploadID	ObjectName
2017-01-13 03:45:26 +0000 CST	15754AF7980C4DFB8193F190837520BB	oss://examplebucket/test.mp4
2017-01-13 03:45:13 +0000 CST	2A1F9B4A95E341BD9285CC42BB950EE0	oss://examplebucket/test.mp4
2017-01-13 03:45:01 +0000 CST	3998971ACAF94AD9AC48EAC1988BE863	oss://examplebucket/test.mp4
2017-01-20 11:16:21 +0800 CST	A20157A7B2FEC4670626DAE0F4C0073C	oss://examplebucket/object.exe

UploadId Number is:4
0.791289(s) elapsed

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要列举另一个阿里云账号下，华东1（杭州）名为test的Bucket内所有文件，命令如下：

```
./ossutil64 ls oss://test -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.24. mb（创建存储空间）

存储空间（Bucket）是用于存储对象（Object）的容器。在上传任意类型的Object前，您需要先创建Bucket。本文介绍如何通过mb命令创建Bucket。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

命令格式

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

参数说明如下：

参数	说明
bucketname	创建的Bucket名称。Bucket名称在OSS范围内必须全局唯一，一旦创建完成则无法修改。
--acl	Bucket的读写权限ACL。取值如下： ● <i>private</i>（默认值）：只有该Bucket的拥有者可以对该Bucket内的文件进行读写操作，其他人无法访问该Bucket内的文件。 ● <i>public-read</i>：只有Bucket拥有者可以对该Bucket内的文件进行写操作，其他用户（包括匿名访问者）都可以对该Bucket中的文件进行读操作。这有可能造成您数据的外泄以及费用激增，若被人恶意写入违法信息还可能会侵害您的合法权益。除特殊场景外，不建议您配置公共读写权限。 ● <i>public-read-write</i>：任何人（包括匿名访问者）都可以对该Bucket内文件进行读写操作。这有可能造成您数据的外泄以及费用激增，请谨慎操作。

参数	说明
<code>--storage-class</code>	Bucket的存储类型。取值如下： • <i>Standard</i>（默认值）：支持频繁的数据访问。 • <i>IA</i>：适用于较低访问频率（平均每月访问频率1到2次）的业务场景，有最低存储时间（30天）和最小计量单位（64 KB）要求。支持数据实时访问，访问数据时会产生数据取回费用。 • <i>Archive</i>：适用于数据长期保存的业务场景，有最低存储时间（60天）和最小计量单位（64 KB）要求。数据需解冻（约1分钟）后访问，解冻会产生数据取回费用。 • <i>ColdArchive</i>：适用于需要超长时间存放的极冷数据，有最低存储时间（180天）和最小计量单位（64 KB）要求。数据需解冻后访问，解冻时间根据数据大小和选择的解冻模式决定，解冻会产生数据取回费用。  说明 在OSS ON云盒使用场景中，仅支持Standard类型。 关于存储类型的更多信息，请参见存储类型介绍。
<code>--redundancy-type</code>	Bucket的数据容灾类型。取值如下： • <i>LRS</i>（默认值）：本地冗余LRS将您的数据冗余存储在同一个可用区的不同存储设备上，可支持两个存储设备并发损坏时，仍维持数据不丢失，可正常访问。 • <i>ZRS</i>：同城冗余ZRS采用多可用区（AZ）机制，将您的数据冗余存储在同一地域（Region）的3个可用区。可支持单个可用区（机房）整体故障时（如断电、火灾等），仍然能够保障数据的正常访问。  注意 • 仅华南1（深圳）、华北2（北京）、华东1（杭州）、华东2（上海）、中国（香港）、新加坡地域支持开启同城冗余存储。 • 在OSS ON云盒使用场景中，不支持使用此选项。

使用示例

- 仅创建examplebucket。

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

如果创建Bucket时未指定Bucket所在地域，则默认在ossutil配置文件中Endpoint指向的地域创建Bucket。例如配置文件中的Endpoint为 `https://oss-cn-hangzhou.aliyuncs.com`，则表示在华东1（杭州）地域创建了Bucket。

- 创建examplebucket，并指定读写权限ACL为私有、存储类型为低频访问以及数据容灾类型为同城冗余ZRS。

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

- 以下输出结果表明已成功创建符合指定条件的Bucket。

```
0.335189(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东2（上海）地域创建名为examplebucket的存储空间，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.25. mkdir（创建目录）

当您希望按实际业务场景对上传至存储空间（Bucket）下的文件（Object）进行合理归类时，您需要先创建目录，然后将目标文件存放至指定目录。本文介绍如何使用mkdir命令创建目录。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
bucketname	目标Bucket名称。
dirname	创建的目录名称。目录名称须以正斜线（/）结尾。若未添加正斜线（/），ossutil会在目录末尾自动添加。
--encoding-type	对 oss://bucket_name 后面的key（目录名称）进行编码，取值为url。如果不指定该选项，则表示目录名称未经过编码。

使用示例

按如下步骤将目标文件上传至指定目录。

1. 创建目录。

o 创建单级目录

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

以下输出结果表明已在目标存储空间examplebucket下成功创建名为dir/的目录。

```
0.385877(s) elapsed
```

o 创建多级目录

当您需要对文件存放的目录进行更精细的分类时，您需要创建多级目录对文件进行管理。例如您需要在目标存储空间examplebucket中Photo/目录下存放2021年份的快照信息。

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

如果误删除了2021/目录，且上一级目录Photo/下文件个数为0，则Photo/目录也会被自动移除。

2. 将文件上传至目标目录

将exampleobject.txt文件上传至存储空间examplebucket下已创建的dir/目录。

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

以上输出结果表明文件已上传至目标目录。

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

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要在另一个阿里云账号下的华东1（杭州）地域名为examplebucket的存储空间下创建目录dir/，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.26. object-tagging（对象标签）

OSS支持以标签的方式对存储的对象（Object）进行分类，方便您管理拥有相同标签的Object，例如通过生命周期规则为相同标签的Object指定过期天数或转换其存储类型。**object-tagging**命令用于添加、修改、获取和删除对象标签。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于对象标签的更多信息，请参见[对象标签](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-idx选项。关于这三个选项的具体用法，请参见[通用选项](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
bucketname	Bucket名称。
prefix	Bucket下的资源，例如目录、文件等。
key	对象标签使用一组键值对（Key-Value）标记对象。单个文件可设置最多10个标签，Key不可重复。设置Key时，需满足以下条件： • 每个Key长度不超过128字符，且区分大小写。 • Key的合法字符集包括大小写字母、数字、空格和以下符号： +=._:/
value	设置Value时，需满足以下条件： • 每个Value长度不超过256字符，且区分大小写。 • Value的合法字符集包括大小写字母、数字、空格和以下符号： +=._:/
--method	请求类型。取值如下： • <i>put</i>：添加或修改对象标签。 • <i>get</i>：获取对象标签。 • <i>delete</i>：删除对象标签。
--encoding-type	对 <code>oss://bucket_name</code> 之后的prefix进行编码，取值为 <i>url</i> 。如果不指定该选项，则表示prefix未经过编码。
-r, --recursive	如果指定该选项时，ossutil将为Bucket下所有符合prefix条件的Object设置标签。如果不指定该选项，则ossutil只为指定Object设置标签。
--version-id	Object的指定版本。仅适用于已开启或暂停版本控制状态Bucket下的Object。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <i>requester</i> 。

添加或修改Object标签

只有Bucket拥有者以及被授予Put Object Tagging的RAM用户拥有设置或修改Object标签的权限。

添加或修改Object标签示例如下：

 **注意** 若Object未设置标签，执行如下操作将为Object添加指定的标签；若Object已配置标签，执行如下操作将覆盖Object原有标签。

- 为目标存储空间examplebucket下的exampleobject.txt文件设置key为tagkey，value为tagvalue的标签信息。

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

- 为目标存储空间examplebucket下的exampleobject.png文件设置两组标签信息。其中一组标签信息为tagkey1#tagvalue1，另一组标签信息为tagkey2#tagvalue2。

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

- 为目标存储空间examplebucket下与前缀test匹配的多个文件设置三组标签信息，标签信息分别为tagkey3#tagvalue3、tagkey4#tagvalue4和tagkey5#tagvalue5。

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

- 为已开启版本控制的存储空间examplebucket下的exampleobject.txt的指定版本设置key为tagkey6，value为tagvalue6的标签信息。

```
./ossutil64 object-tagging --method put oss://examplebucket/exampleobject.txt tagkey6#tagvalue6 --version-id CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5ZjNhYzQzZjNlYTAYZDE3****
```

关于获取Object所有版本的具体操作，请参见[ls（列举账号级别下的资源）](#)。

- 以上示例操作成功后，返回结果中将包含设置标签信息所用时长，示例如下：

```
0.106852(s) elapsed
```

获取Object标签信息

只有Bucket拥有者以及被授予Get Object Tagging的RAM用户拥有获取Object标签信息的权限。

获取Object标签信息的示例如下：

- 获取单个Object的标签信息

获取目标存储空间examplebucket下exampleobject.txt的标签信息。

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

以下输出结果表明已成功获取exampleobject.txt的标签信息，标签信息的key为tagkey，value为tagvalue。

object index	tag index	tag key	tag value	object
1	0	"tagkey"	"tagvalue"	oss://examplebucket/exampleobject.txt

0.068156(s) elapsed

- 获取多个Object的标签信息

获取目标存储空间examplebucket下与前缀test匹配的文件标签信息。


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

以下输出结果表明已成功获取与前缀test匹配的所有文件的标签信息，分别为tagkey3#tagvalue3、tagkey4#tagvalue4和tagkey5#tagvalue5。

object index	tag index	tag key	tag value	object
1	0	"tagkey3"	"tagvalue3"	oss://examplebucket/test
1	1	"tagkey4"	"tagvalue4"	oss://examplebucket/test
1	2	"tagkey5"	"tagvalue5"	oss://examplebucket/test

0.093040(s) elapsed

删除Object标签信息

只有Bucket拥有者以及被授予DeleteObjectTagging的RAM用户拥有删除Object标签信息的权限。

删除Object标签信息的示例如下：

- 删除单个Object的标签信息

删除目标存储空间examplebucket下exampleobject.txt的标签信息。

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

- 删除多个Object的标签信息

删除目标存储空间examplebucket下与前缀test匹配的所有文件的标签信息。

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

- 以上示例操作成功后，返回结果中将包含删除标签信息所用时长，示例如下：

```
0.148970(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东2（上海）地域下目标存储空间testbucket下的exampletest.png文件设置key为tagkey7、value为tagvalue7的标签信息，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.27. probe（探测状态）

probe命令是针对OSS访问的探测命令，可用于排查本地与OSS之间的网络状态、上传下载带宽、本地符号链接（软链接）状态等。

 **注意**

本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

使用说明

当您需要了解本地与OSS之间的网络异常、查看上传下载带宽、上传大量软链接文件时，都可以使用probe进行探测。

- 探测网络异常

ossutil通过上传或下载的方式探测本地与OSS之间的网络状况。如果您在探测的同时需要将指定文件上传或下载到指定位置，建议您使用实际的文件进行上传下载探测；若您仅需要探测网络异常，可以在上传或下载探测时不添加文件名，ossutil会使用临时文件进行探测，并在探测结束后删除这些文件。
- 查看上传下载带宽

探测上传下载带宽时，OSS会根据您的设备CPU及您的带宽给出一个上传下载的并发建议值。您可以按照ossutil的建议设置并发，以最大限度地利用您的带宽。
- 探测本地软链接状态

当您上传大量软链接文件时，若某个链接出现异常会导致上传中断。建议您在上传前，先探测本地软链接文件是否存在异常。若存在异常，请修复异常文件后再进行批量上传。

probe命令运行后，您可以查看到任务执行的步骤及结果。

- 上传和下载探测
 - 执行步骤出现✓表示本项检测通过，×表示未通过。
 - 若探测成功，ossutil会给出文件的大小、上传下载时间、文件路径等信息。
 - 若探测失败，ossutil会给出导致错误的原因或错误码，您可以根据提示排查问题原因。

关于错误码的介绍，请参见[错误响应](#)。
 - 探测结束后会在ossutil的安装目录生成一个格式为 `logOssProbe+探测结束时间.log` 的日志文件，里面包含此次probe命令执行的详细信息。
- 指定项目探测

探测上传下载带宽、本地软链接状态时，ossutil会直接给出探测结果和可能的建议。

上传文件并输出探测报告

ossutil上传文件到目标存储空间（Bucket）的方式探测本地和目标Bucket之间的网络状态。

● 命令格式

```
./ossutil64 probe [--upload [file_name]] [--bucketname bucket_name] [--object object_name] [--addr domain_name] [--upmode]
```

参数说明如下：

参数	是否必选	说明
--upload	是	指定探测方式为上传探测。
file_name	否	填写要上传至目标Bucket的本地文件完整路径；置空此项时，ossutil会生成一个临时文件进行上传探测。

参数	是否必选	说明
--bucketname	否	目标Bucket名称。
--object	否	增加此项并指定文件名称，ossutil会将上传文件按指定名称保存在Bucket中；若不增加此项，ossutil会在探测结束后删除上传的文件。
--addr	否	增加此项并填写正确的网络地址，ossutil会通过ping操作验证本地到目标地址的网络连通性。 默认值： <code>www.aliyun.com</code>
--upmode	否	指定文件的上传方式。 取值： - normal（默认值）：简单上传 - append：追加上传 - multipart：分片上传

- 使用示例

- 上传随机文件并指定ping操作的目标地址

上传的目标Bucket名为examplebucket，ping操作的目标网络地址为 `aliyun.com`。命令如下：

```
./ossutil64 probe --upload --bucketname examplebucket --addr aliyun.com
```

输出如下：

```
begin parse parameters and prepare file...[✓]
begin network detection...[✓]
begin upload file(normal)...[✓]
***** upload result *****
upload file:success
upload file size:122880(byte)
upload time consuming:245(ms)
(only the time consumed by probe command)
***** report log info*****
report log file:/root/logOssProbe20201201173031.log
```

- 使用默认上传方式上传指定文件，并在探测结束后删除该文件

将本地根目录下的文件 *example.txt* 上传到名为 *examplebucket* 的 Bucket。命令如下：

```
./ossutil64 probe --upload example.txt --bucketname examplebucket
```

输出如下：

```
begin parse parameters and prepare file...[✓]
begin network detection...[✓]
begin upload file(normal)...[✓]
***** upload result *****
upload file:success
upload file size:7(byte)
upload time consuming:224(ms)
(only the time consumed by probe command)
upload object is example.txt
***** report log info*****
report log file:/root/logOssProbe20201201173841.log
```

- 使用追加上传方式上传指定文件，并在探测结束后保留该文件

将本地根目录下的文件 *example.txt* 上传到名为 *examplebucket* 的 Bucket。命令如下：

```
./ossutil64 probe --upload example.txt --bucketname examplebucket --object example.txt
--upmode append
```

输出如下：

```
begin parse parameters and prepare file...[✓]
begin network detection...[✓]
begin upload file(append)...[✓]
***** upload result *****
upload file:success
upload file size:7(byte)
upload time consuming:171(ms)
(only the time consumed by probe command)
upload object is example.txt
***** report log info*****
report log file:/root/logOssProbe20201201174126.log
```

通过文件URL下载文件并输出探测报告

ossutil通过使用文件URL下载目标文件到本地的方式探测本地和目标Bucket之间的网络状态。

- 命令格式

```
./ossutil64 probe [--download] [--url http_url] [--addr=domain_name] [file_name]
```

参数说明如下：

参数	是否必选	说明
--download	是	指定探测方式为下载探测。

参数	是否必选	说明
--url	是	填写文件URL，ossutil会将URL对应的文件下载到本地。 - 公共读文件：直接填写文件URL。例如 <code>https://examplebucket.oss-cn-beijing.aliyuncs.com/example.jpg</code>。 - 私有文件：填写带签名的文件URL，URL需使用双引号（<code>"</code>）包裹。例如 <code>"https://examplebucket.oss-cn-beijing.aliyuncs.com/example.jpg?Expires=1552015472&OSSAccessKeyId=TMP.*****5r9f1FV12y8_Qis6LUVmvoSCUSs7aboCCHtydQ0axN32Sn-UvyY3AAAwLAIUarYNLcO87AKMEcE5O3A*****oCFAQuRdZYyVFYqOW8QkGAN-bamUiQ&Signature=bIa41lbMbldrl7rwckr%2FXXvTtxw%3D"</code>。
--addr	否	增加此项并填写正确的网络地址，ossutil会通过ping操作验证本地到目标地址的网络连通性。 默认值：<code>www.aliyun.com</code>
file_name	否	设置文件下载到本地后的保存路径。 - 仅指定文件名，未指定目录：ossutil会将文件按指定名称保存到ossutil的安装目录。 - 仅指定目录，未指定文件名：ossutil会将文件按原名称保存到指定目录。 - 置空此项：ossutil会将文件按原名称保存到ossutil的安装目录。

● 使用示例

○ 通过文件URL下载目标文件并重命名

文件URL为 `https://examplebucket.oss-cn-beijing.aliyuncs.com/example.txt`，指向examplebucket中一个公共读文件 `example.txt`，下载到本地后重命名为 `/localfile/test.txt`。命令如下：

```
./ossutil64 probe --download --url https://examplebucket.oss-cn-beijing.aliyuncs.com/example.txt /localfile/test.txt
```

输出如下：

```
begin parse parameters and prepare object...[✓]
begin network detection...[✓]
begin download file...[✓]
***** download result *****
download file:succcess
download file size:57374182(byte)
download time consuming:1246(ms)
(only the time consumed by probe command)
download file is /localfile/test.txt
***** report log info*****
report log file:/root/logOssProbe20201202171639.log
```

- 通过文件URL下载目标文件并指定ping操作的目标地址

文件URL为 `https://examplebucket.oss-cn-beijing.aliyuncs.com/example.txt` , 指向examplebucket中一个公共读文件 `example.txt` , ping操作的目标地址为 `aliyun.com` 。命令如下:

```
./ossutil64 probe --download --url https://examplebucket.oss-cn-beijing.aliyuncs.com/example.txt --addr aliyun.com
```

输出如下:

```
begin parse parameters and prepare object...[✓]
begin network detection...[✓]
begin download file...[✓]
***** download result *****
download file:success
download file size:57374182(byte)
download time consuming:1344(ms)
(only the time consumed by probe command)
download file is /root/example.txt
***** report log info*****
report log file:/root/logOssProbe20201202172232.log
```

直接下载指定文件并输出探测报告

ossutil通过直接下载目标Bucket中的文件到本地的方式探测本地和目标Bucket之间的网络状态。

- 命令格式

```
./ossutil64 probe [--download] [--bucketname bucket_name] [--object object_name] [--addr domain_name] [file_name]
```

参数说明如下:

参数	是否必选	说明
--download	是	指定探测方式为下载探测。
--bucketname	是	目标Bucket名称。
--object	否	增加此项并指定文件名称, ossutil会将指定文件下载到本地; 置空此项, ossutil会生成一个临时文件上传到目标Bucket后再将其下载, 探测结束后会将该临时文件从目标Bucket中删除。
--addr	否	增加此项并填写正确的网络地址, ossutil会通过ping操作验证本地到目标地址的网络连通性。 默认值: <code>www.aliyun.com</code>

参数	是否必选	说明
file_name	否	设置文件下载到本地后的保存路径。 - 仅指定文件名，未指定目录：ossutil会将文件按指定名称保存到ossutil的安装目录。 - 仅指定目录，未指定文件名：ossutil会将文件按原名称保存到指定目录。 - 置空此项：ossutil会将文件按原名称保存到ossutil的安装目录。

● 使用示例

○ 下载指定文件并重命名

将名为examplebucket的Bucket中的文件 `/ossfolder/example.txt` 下载到本地后重命名为 `/localfolder/text.txt`。

```
./ossutil64 probe --download --bucketname examplebucket --object /ossfolder/example.txt /localfolder/text.txt
```

输出如下：

```
begin parse parameters and prepare object...[✓]
begin network detection...[✓]
begin download file...[✓]
***** download result *****
download file:success
download file size:57374182(byte)
download time consuming:1108(ms)
(only the time consumed by probe command)
download file is /localfolder/text.txt
***** report log info*****
report log file:/root/logOssProbe20201202173032.log
```

○ 下载临时文件并指定ping操作的目标地址

目标Bucket名为examplebucket，ping操作的目标地址为 `aliyun.com`。命令如下：

```
./ossutil64 probe --download --bucketname examplebucket --addr aliyun.com
```

输出如下：

```
begin parse parameters and prepare object...[✓]
begin network detection...[✓]
begin download file...[✓]
***** download result *****
download file:success
download file size:122880(byte)
download time consuming:516(ms)
(only the time consumed by probe command)
download file is /root/oss-test-probe-1606902911701892000-3ifwj9t0ln
***** report log info*****
report log file:/root/logOssProbe20201202173512.log
```

指定项目探测

ossutil可通过指定探测项目的方式探测本地软链接文件状态、上传和下载带宽，并根据结果给出异常软链接的名称、上传下载并发数建议。

命令格式

```
./ossutil64 probe {--probe-item item_value} {--bucketname bucket-name} [--object object_name]
```

参数说明如下：

参数	是否必选	说明
--probe-item	是	指定探测项目。 取值： - cycle-symlink：探测本地是否存在异常软链接。 - upload-speed：探测上传带宽。 - download-speed：探测下载带宽。
--bucketname	当--probe-item取值不为 cycle-symlink 时必选	目标Bucket名称。
--object	当--probe-item取值为 do wnload-speed 时必选	填写目标文件的访问路径。目标文件需真实存在且建议大于5 MB。例如 ossfolder/example.txt 。

使用示例

- 探测本地根目录下localfolder目录是否存在异常软链接

命令如下：

```
./ossutil64 probe --probe-item cycle-symlink /root/localfolder
```

输出如下：

```
Error: stat /root/localfolder/example.jpg: no such file or directory
```

输出内容表示example.jpg为异常软链接。

◦ 探测上传带宽

上传一个临时文件到examplebucket，并根据当前设备的硬件配置及上传带宽给出上传并发数的配置建议。

命令如下：

```
./ossutil64 probe --probe-item upload-speed --bucketname examplebucket
```

输出如下：

```
cpu core count:2
parallel:2,average speed:679.72(KB/s),current speed:1344.00(KB/s),max speed:1440.00(KB/s))
parallel:3,average speed:643.31(KB/s),current speed:704.00(KB/s),max speed:1632.00(KB/s))
parallel:4,average speed:646.62(KB/s),current speed:512.00(KB/s),max speed:1600.00(KB/s))
suggest parallel is 2, max average speed is 679.72(KB/s)
```

输出内容表示设备CPU为双核，上传最大平均带宽为679.72 KB/s，建议上传并发数设置为2。

◦ 探测下载带宽

将examplebucket中的文件 `example.txt` 下载到本地，并根据当前设备的硬件配置及下载带宽给出下载并发数的配置建议。

命令如下：

```
./ossutil64 probe --probe-item download-speed --bucketname examplebucket --object example.txt
```

输出如下：

```
cpu core count:2
parallel:2,average speed:12524.93(KB/s),current speed:12288.63(KB/s),max speed:14302.25(KB/s)
parallel:3,average speed:12564.45(KB/s),current speed:12144.39(KB/s),max speed:14484.24(KB/s)
parallel:4,average speed:12545.21(KB/s),current speed:12766.58(KB/s),max speed:13534.42(KB/s)
suggest parallel is 3, max average speed is 12564.45(KB/s)
```

输出内容表示设备CPU为双核，下载最大平均带宽为12564.45 KB/s，建议下载并发数设置为3。

2.3.28. read-symlink（读取软链接）

`read-symlink`命令用于读取软链接文件的描述信息，包括软链接文件的ETag值、最后更新时间等。此操作要求用户对软链接文件有读权限。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的`./ossutil64`替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。

- ii. 在本文已有示例的基础上添加 `--sign-version`、`--region`以及 `--cloudbox-id`选项。关于这三个选项的具体用法，请参见[通用选项](#)。

命令格式

```
./ossutil64 read-symlink oss://bucketname/objectname [--encoding-type <value>] [--payer <value>]
```

参数及选项说明如下：

配置项	说明
bucketname	Bucket名称。
objectname	软链接文件名称。
--encoding-type	对软链接文件名称进行编码，取值为 <i>url</i> 。如果不指定该选项，则表示文件名称未经过编码。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <i>requester</i> 。

使用示例

以下示例用于读取目标存储空间examplebucket下软链接文件test.jpg的描述信息。

```
./ossutil64 read-symlink oss://examplebucket/test.jpg
```

以下输出结果表明已成功获取软链接文件test.jpg的ETag值、最后更新时间（Last-Modified）以及指向的目标文件（X-Oss-Symlink-Target）为example.jpg。

```
Etag           : 938F26218CE422CBEEE0B6543A2B2D
Last-Modified  : 2021-04-21 18:00:13 +0800 CST
X-Oss-Symlink-Target : example.jpg
0.217317(s) elapsed
```

如果此命令操作的文件类型不是软链接文件，将返回错误 `NotSymlink` 。

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要获取另一个阿里云账号下，华东2（上海）地域下目标存储空间testbucket中名为testobject.png软链接文件的信息，命令如下：

```
./ossutil64 read-symlink oss://testbucket/testobject.png -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.29. referer（防盗链）

对象存储OSS支持对存储空间（Bucket）设置防盗链，即通过对访问来源设置白名单的机制，避免OSS资源被其他人盗用。**referer**命令用于添加、修改、查询、删除Bucket的防盗链配置。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于防盗链的更多信息，请参见[防盗链](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

添加或修改防盗链配置

如果Bucket此前未添加过防盗链配置，则使用该命令将为Bucket添加防盗链配置。如果Bucket此前已添加防盗链配置，则使用该命令将直接覆盖已有的防盗链配置。

添加或修改防盗链配置的命令格式和示例如下：

- 命令格式

```
./ossutil64 referer --method put oss://bucketname refererconfig [--disable-empty-referer]
```

参数及选项说明如下：

配置项	说明
bucketname	待添加或修改防盗链配置的Bucket名称。
refererconfig	填写域名或IP地址，支持通配符星号（*）和问号（?），多个Referer需以空格分隔。配置示例如下： - 配置为 www.aliyun.com 时，可匹配www.aliyun.com/123、www.aliyun.com.cn等以www.aliyun.com为前缀的地址。 - 配置为 *www.aliyun.com/，可匹配http://www.aliyun.com/和https://www.aliyun.com/的地址。 - 通配符星号（*）表示使用星号代替0个或多个字符。例如配置为 *.aliyun.com，可匹配help.aliyun.com、www.aliyun.com等地址。 - 通配符问号（?）表示使用问号代替一个字符。 - 支持带端口的域名或IP，例如www.example.com:8080、10.0.0.0:8080等地址。
--disable-empty-referer	选择是否允许空Referer。 - 不添加此选项时，表示允许空Referer，即HTTP或HTTPS Header中不带Referer字段或Referer字段为空。 - 添加此选项后，表示不允许空Referer，即HTTP或HTTPS Header中包含Referer字段的请求才能访问OSS资源。

● 使用示例

为目标存储空间examplebucket设置防盗链，仅HTTP或HTTPS Header中与 `*www.aliyun.com` 匹配的请求允许访问examplebucket，且不允许空Referer。

```
./ossutil64 referer --method put oss://examplebucket *www.aliyun.com --disable-empty-referer
```

为目标存储空间examplebucket设置防盗链，仅HTTP或HTTPS Header中包含 `example.aliyundoc.com` 的请求允许访问examplebucket，且允许空Referer。

```
./ossutil64 referer --method put oss://examplebucket example.aliyundoc.com
```

以下输出结果表明已成功添加防盗链配置。

```
0.134839(s) elapsed
```

查看防盗链配置

● 命令格式

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

参数说明如下：

参数	说明
bucketname	待获取防盗链配置的Bucket名称。
local_xml_file	存放防盗链配置的本地文件名称，例如 <code>localfile.txt</code> 。如果未指定此参数，则防盗链配置将直接输出到屏幕。

● 使用示例

- 获取存储空间examplebucket的防盗链配置，并将配置结果写入本地文件 `localfile.txt` 。

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

以下输出结果表明已成功获取防盗链配置。

```
0.212407(s) elapsed
```

- 获取存储空间examplebucket的防盗链配置，并将配置结果直接输出屏幕。

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

以下输出结果表明examplebucket的防盗链配置中，仅HTTP或HTTPS Header中与 *www.aliyun.com 匹配的请求允许访问examplebucket，且不允许空Referer。

```
<?xml version="1.0" encoding="UTF-8"?>
  <RefererConfiguration>
    <AllowEmptyReferer>false</AllowEmptyReferer>
    <RefererList>
      <Referer>*www.aliyun.com</Referer>
    </RefererList>
  </RefererConfiguration>
0.080482(s) elapsed
```

删除防盗链配置

- 命令格式

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

bucketname表示待删除防盗链配置的Bucket名称。

- 使用示例

删除存储空间examplebucket的防盗链配置。

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

以下输出结果表明已成功删除防盗链配置。

```
0.212409(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）地域下名为testbucket的存储空间设置防盗链，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.30. replication（跨区域复制）

跨区域复制（Cross-Region Replication）是跨不同OSS数据中心（地域）的存储空间（Bucket）自动、异步（近实时）复制文件（Object）。您可以通过replication命令管理跨区域复制规则配置。规则配置完成后，OSS会根据指定规则将源Bucket内的Object复制到不同区域的目标Bucket。

 注意

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于跨区域复制的使用场景及注意事项的更多信息，请参见[跨区域复制](#)。

添加跨区域复制规则

您可以通过跨区域复制规则将源Bucket中的数据同步到多个目标Bucket。单个Bucket关联的复制规则数量不能超过100条。

要添加跨区域复制规则，首先要创建本地文件并在文件中添加跨区域复制规则，然后将跨区域复制规则添加到源Bucket。

- 命令格式

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

参数说明如下：

参数	说明
bucketname	待添加跨区域复制规则的源Bucket名称。
local_xml_file	配置跨区域复制规则的本地文件名称，例如 localfile.txt 。

- 使用示例

- i. 在本地创建名为 localfile.txt 文件，并根据使用场景配置跨区域复制规则。

将源Bucket内与指定前缀srcdir匹配的文件的增、删、改操作同步到华东1（杭州）地域下名为destbucket的目标Bucket。此外，指定进行跨区域复制时开启传输加速，且同步历史数据。

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Rule>
    <PrefixSet>
      <Prefix>srcdir</Prefix>
    </PrefixSet>
    <Action>ALL,PUT</Action>
    <Destination>
      <Bucket>destbucket</Bucket>
      <Location>oss-cn-hangzhou</Location>
      <TransferType>oss_acc</TransferType>
    </Destination>
    <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
  </Rule>
</ReplicationConfiguration>
```

- ii. 为源存储空间srcbucket添加跨区域复制规则。

```
./ossutil64 replication --method put oss://srcbucket localfile.txt
```

- 返回结果

以下输出结果表明已为srcbucket成功添加跨区域复制规则。

```
0.856895(s) elapsed
```

查看跨区域复制配置

- 命令格式

```
./ossutil64 replication --method get oss://bucketname
```

- 使用示例

获取存储空间srcbucket的跨区域复制配置，并将配置结果直接输出屏幕。

```
./ossutil64 replication --method get oss://srcbucket
```

以下输出结果表明已成功获取srcbucket的跨区域复制配置。

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Rule>
    <ID>37417af4-f2dc-4b24-92d3-82092af6****</ID>
    <Action>ALL</Action>
    <Destination>
      <Bucket>destbucket</Bucket>
      <Location>oss-cn-beijing</Location>
    </Destination>
    <Status>doing</Status>
    <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
  </Rule>
</ReplicationConfiguration>
0.069195(s) elapsed
```

查询可同步的目标地域

- 命令格式

```
./ossutil64 replication --method get --item location oss://bucketname
```

- 使用示例

查询源存储空间srcbucket的数据可同步到哪些目标地域。

```
./ossutil64 replication --method get --item location oss://srcbucket
```

以下输出结果表明已成功获取可同步到的目地域。

```
<?xml version="1.0" ?>
<ReplicationLocation>
  <Location>oss-cn-beijing</Location>
  <Location>oss-cn-qingdao</Location>
  <Location>oss-cn-shenzhen</Location>
  <Location>oss-cn-hongkong</Location>
  <Location>oss-us-west-1</Location>
  <LocationTransferTypeConstraint>
    <LocationTransferType>
      <Location>oss-cn-hongkong</Location>
      <TransferTypes>
        <Type>oss_acc</Type>
      </TransferTypes>
    </LocationTransferType>
    <LocationTransferType>
      <Location>oss-us-west-1</Location>
      <TransferTypes>
        <Type>oss_acc</Type>
      </TransferTypes>
    </LocationTransferType>
  </LocationTransferTypeConstraint>
</ReplicationLocation>
0.226523(s) elapsed
```

查询指定Bucket的跨区域复制进度

● 命令格式

```
./ossutil64 replication --method get --item progress oss://bucketname [ruleID]
```

参数及选项说明如下：

参数	说明
bucketname	待获取跨区域复制进度的Bucket名称。
ruleID	查询指定ruleID对应的跨区域复制进度。如果不指定ruleID，则查询Bucket所有关联的跨区域复制进度。目前只支持将跨区域复制进度输出到屏幕上。

● 使用示例

获取srcbucket指定ruleID对应的跨区域复制进度。

```
./ossutil64 replication --method get --item progress oss://srcbucket 37417af4-f2dc-4b24-92d3-82092af6****
```

获取srcbucket所有关联规则的跨区域复制进度。

```
./ossutil64 replication --method get --item progress oss://srcbucket
```

以下输出结果表明源存储空间srcbucket在开启跨区域复制前已有的数据复制到目标Bucket的进度为100%，在开启跨区域复制后，早于时间点（2021-08-09T06:00:59.000Z）写入srcbucket中的数据都已复制到目标存储空间destbucket。


```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationProgress>
  <Rule>
    <ID>37417af4-f2dc-4b24-92d3-82092af6****</ID>
    <Action>ALL</Action>
    <Destination>
      <Bucket>destbucket</Bucket>
      <Location>oss-cn-beijing</Location>
    </Destination>
    <Status>doing</Status>
    <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
    <Progress>
      <HistoricalObject>1.00</HistoricalObject>
      <NewObject>2021-08-09T06:00:59.000Z</NewObject>
    </Progress>
  </Rule>
</ReplicationProgress>
0.125002(s) elapsed
```

删除跨区域复制配置

- 命令格式

```
./ossutil64 replication --method delete oss://bucketname ruleID
```

- 使用示例

删除srcbucket指定ruleID关联的跨区域复制配置。

```
./ossutil64 replication --method delete oss://srcbucket 37417af4-f2dc-4b24-92d3-82092af6*
***
```

以下输出结果表明已成功删除指定ruleID关联的跨区域复制配置。

```
0.069195(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东2（上海）地域下的testbucket配置跨区域复制规则，跨区域复制规则通过本地文件localfile.txt进行配置。命令如下：

```
./ossutil64 replication --method put oss://testbucket localfile.txt -e oss-cn-shanghai.aliy
uncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.31. request-payment（请求者付费）

请求者付费模式是指由请求者支付访问存储空间（Bucket）内数据时产生的费用，而Bucket拥有者仅支付存储费用。当您希望共享数据，但又不希望支付因共享数据产生的额外费用时，您可以使用`request-payment`命令设置请求者付费模式。

 注意

本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的`./ossutil64`替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

关于请求者付费模式的更多信息，请参见[请求者付费模式](#)。

设置请求者付费模式

命令格式

```
./ossutil64 request-payment --method put oss://bucketname payment
```

参数说明如下：

参数	说明
bucketname	填写待设置请求者付费模式的目标Bucket名称。
payment	第三方用户访问目标Bucket内的数据产生费用时的付费模式。 取值如下： <i>Requester</i>: 访问此Bucket内的数据产生的所有费用由请求者支付。 启用请求者付费模式后，不允许匿名访问此Bucket。请求方必须提供身份验证信息，以便OSS能够识别请求方，从而对请求方而非Bucket拥有者收取请求所产生的费用。当请求者是通过扮演阿里云RAM角色来请求数据时，该角色所属的账户将为此请求付费。 <i>BucketOwner</i>: 访问此Bucket内资源产生的所有费用由Bucket拥有者支付。

使用示例

为目标存储空间examplebucket设置请求者付费模式。

```
./ossutil64 request-payment --method put oss://examplebucket Requester
```

为目标存储空间examplebucket设置Bucket拥有者付费模式。

```
./ossutil64 request-payment --method put oss://examplebucket BucketOwner
```

以下输出结果表明已成功设置请求付费模式。

```
0.106852(s) elapsed
```

获取请求者付费模式

命令格式

```
./ossutil64 request-payment --method get oss://bucketname
```

bucketname填写待获取请求者付费模式的目标Bucket。

- 使用示例

获取目标存储空间examplebucket的请求者付费模式。

```
./ossutil64 request-payment --method get oss://examplebucket
```

以下输出结果表明examplebucket已开启请求者付费模式。

```
Requester
0.072024(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东2（上海）地域名为testbucket的存储空间开启请求者付费模式，命令如下：

```
./ossutil64 request-payment --method put oss://testbucket -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.32. restore（解冻文件）

访问归档或冷归档类型文件（Object）之前，需要通过restore命令解冻文件。

注意

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于归档或冷归档文件的解冻状态以及费用说明，请参见[解冻文件](#)。

命令格式

```
./ossutil64 restore oss://bucketname[/prefix] [local_xml_file]
[--encoding-type <value>]
[--payer <value>]
[--version-id <value>]
[-r, --recursive]
[-f, --force]
[--object-file, <value>]
[--snapshot-path, <value>]
[--disable-ignore-error]
[--retry-times <value>]
[-j, --job <value>]
```

参数及选项说明如下：

配置项	说明
bucketname	Bucket名称。
prefix	Bucket下的资源，例如目录、文件等。
local_xml_file	本地XML格式文件，用于保存冷归档类型文件的解冻参数。
--encoding-type	对prefix名称进行编码，取值为url。如果不指定该选项，则表示prefix未经过编码。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为requester。
--version-id	Object的指定版本。仅适用于已开启或暂停版本控制状态Bucket下的Object。
-r, --recursive	如果指定该选项时，ossutil将解冻所有与prefix匹配的文件。如果不指定该选项，则ossutil只解冻指定文件。
-f, --force	强制操作，不进行询问提示。
--object-file	该选项用于批量解决多个归档或冷归档类型Object。使用该选项的步骤如下： 1. 您需要指定.txt或者XML格式的本地文件，并在本地文件中换行填写所有待解冻的Object。 2. ossutil读取本地文件中的所有Object，然后对这些Object进行批量解冻。  说明 当其中一个Object操作出现错误时，ossutil会将该Object的错误信息记录到report文件，并继续解冻其他Object。成功操作的Object信息不会被记录到report文件中。
--snapshot-path	如果指定该选项，则只对本次操作的Object生成快照。如果操作的Object已经存在快照信息，则忽略本次操作。  说明 该选项需结合-r, --recursive或--object-file选项使用。
--disable-ignore-error	批量操作时不忽略错误。
--retry-times	发生错误后的重试次数。默认值为10，取值范围为1~500。
-j, --job	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。

解冻归档类型Object

解冻归档类型Object需要1分钟时间，解冻中的Object不支持读取。

解冻状态默认持续1天。对解冻状态的文件调用restore命令，会将Object的解冻状态延长1天，最多可以延长到7天，之后Object又回到初始时的冷冻状态。

- 解冻单个归档类型Object

- 以下示例用于解冻目标存储空间examplebucket下名为exampleobject.txt的归档类型Object。

```
./ossutil64 restore oss://examplebucket/exampleobject.txt
```

- 以下示例用于解冻目标存储空间examplebucket下归档类型文件exampleobject.txt的指定版本。

```
./ossutil64 restore oss://examplebucket/exampleobject.txt --version-id CAEQARiBgID8rum  
R2hYiIGUyOTAYZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3****
```

关于获取Object所有版本的具体操作，请参见[ls（列举账号级别下的资源）](#)。

- 解冻多个归档类型Object

- 示例一：不同目录下

当您需要解冻examplebucket中的多个归档类型Object，且这些Object归属于Bucket的不同目录下，例如exampleobject1.jpg位于examplebucket根目录下，exampleobject2.png位于examplebucket的dir1/目录下，exampleobject3.txt位于examplebucket的dir2/目录下。您需要按照以下步骤完成解冻操作。

- a. 将待解冻的多个归档类型Object写入本地文件localfile.txt。

```
exampleobject1.jpg  
dir1/exampleobject2.png  
dir2/exampleobject3.txt
```

- b. 解冻多个归档类型Object。

结合--object-file选项解冻目标存储空间examplebucket下的多个冷归档类型Object。

```
./ossutil64 restore oss://examplebucket --object-file localfile.txt --snapshot-path  
dir/
```

- 示例二：相同目录下

- 方法一

您可以参考[示例一](#)解冻某个Bucket中相同目录下的多个归档存储类型Object。

- 方法二

结合-r选项，解冻examplebucket中指定目录dest下的所有归档类型Object。

```
./ossutil64 restore oss://examplebucket/dest -r
```

- 返回结果

以上操作成功后，返回结果中将包含发起解冻请求所用时长，示例如下：

```
0.106852(s) elapsed
```

解决冷归档类型Object



注意 根据解冻Object的大小，实际解冻时间可能会有变化，请以实际解冻时间为准。

解决单个或多个冷归档类型Object前，您需要在本地创建XML格式文件config.xml，并在文件中配置如下解冻参数。

```
<RestoreRequest>
  <Days>3</Days>
  <JobParameters>
    <Tier>Expedited</Tier>
  </JobParameters>
</RestoreRequest>
```

配置参数说明如下：

参数	说明
Days	冷归档文件保持解冻状态的时间，单位为天。 取值范围：1~365
Tier	冷归档文件解冻优先级。 取值如下： • <i>Expedited</i>（高优先级）：1小时内完成解冻。 • <i>Standard</i>（标准）：2~5小时完成解冻。 • <i>Bulk</i>（批量）：5~12小时完成解冻。

● 解冻单个冷归档类型Object

按照 *config.xml* 文件配置，指定在1小时内解冻目标存储空间examplebucket下的exampleobject.jpg，并且解冻状态保持3天。

```
./ossutil64 restore oss://examplebucket/exampleobject.jpg config.xml
```

● 解冻多个冷归档类型Object

○ 示例一：不同目录下

当您需要解冻examplebucket中的多个冷归档类型Object，且这些Object归属于Bucket的不同目录下，例如exampleobject1.jpg位于examplebucket根目录下，exampleobject2.png位于examplebucket的dir1/目录下，exampleobject3.txt位于examplebucket的dir2/目录下。您需要按照以下步骤完成解冻操作。

a. 将待解冻的多个冷归档类型Object写入本地文件localfile.txt。

```
exampleobject1.jpg
dir1/exampleobject2.png
dir2/exampleobject3.txt
```

b. 解冻多个Object。

结合 *--object-file* 选项，按照 *config.xml* 文件配置，指定在1小时内解冻目标存储空间examplebucket下的多个冷归档类型Object，并且解冻状态保持3天。

```
./ossutil64 restore oss://examplebucket --object-file localfile.txt --snapshot-path
dir/ config.xml
```

- 示例二：相同目录下

- 方法一

- 您可以参考[示例一](#)解冻某个Bucket中相同目录下的多个冷归档存储类型Object。

- 方法二

- 结合-r选项，解冻examplebucket下指定目录dest下的所有冷归档类型Object。

```
./ossutil64 restore oss://examplebucket/dest config.xml -r config.xml
```

- 返回结果

以上操作成功后，返回结果中将包含发起解冻请求所用时长，示例如下：

```
0.106852(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要解冻另一个阿里云账号下，华东2（上海）地域下目标存储空间testbucket下的exampletest.png文件，命令如下：

```
./ossutil64 restore oss://testbucket/exampletest.png -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.33. revert-versioning（恢复版本）

revert-versioning命令用于在开启版本控制的存储空间（Bucket）中，将已删除的文件（Object）恢复至最近的版本。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于版本控制的更多信息，请参见[版本控制介绍](#)。
- 关于删除标记的更多信息，请参见[删除标记](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

命令格式

```
./ossutil64 revert-versioning oss://bucketname[/prefix]
[--encoding-type <value>]
[-r, --recursive]
[--start-time <value> ]
[--end-time <value>]
[--include <value>]
[--exclude <value>]
[--payer <value>]
```

参数及选项说明如下：

配置项	说明
bucketname	Bucket名称。
prefix	Bucket下的资源、例如目录、文件等。
--encoding-type	对 <code>oss://bucket_name</code> 之后的prefix进行编码，取值为 <i>url</i> 。如果不指定该选项，则表示prefix未经过编码。
-r, --recursive	如果指定该选项时，ossutil将恢复Bucket下所有符合prefix条件的Object为最近的版本。如果不指定该选项，则ossutil只恢复指定Object为最近的版本。
--start-time	值为Linux或Unix系统下的时间戳。如果输入该选项，则删除时间早于该时间的Object会被忽略。
--end-time	值为Linux或Unix系统下的时间戳。如果输入该选项，则删除时间晚于该时间的Object会被忽略。
--include	包含符合指定条件的所有Object。
--exclude	不包含任何符合指定条件的Object。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <i>requester</i> 。

使用示例

- 恢复示例
 - 将目标存储空间examplebucket下处于删除状态的exampleobject.jpg恢复为最近的版本。

./ossutil64 revert-versioning oss://examplebucket/exampleobject.jpg
 - 结合 -r 选项，将目标存储空间examplebucket下匹配指定前缀destdir且处于删除状态的多个Object恢复为最近的版本。

./ossutil64 revert-versioning oss://examplebucket/destdir -r
 - 结合 -r 选项，将目标存储空间examplebucket内所有处于删除状态的Object恢复为最近的版本。

./ossutil64 revert-versioning oss://examplebucket -r

- 将在指定时间范围内删除的Object恢复为最近的版本

结合 `-t` 选项，将目标存储空间examplebucket下于北京时间2020年6月16日16:22:58至2020年6月16日16:39:38内删除的Object恢复至最近的版本。

```
./ossutil64 revert-versioning oss://examplebucket -r --start-time 1592295778 --end-time 1592296778
```

- 将所有匹配指定条件且已删除的Object恢复为最近的版本

- 结合 `-t` 选项，将目标存储空间examplebucket下所有格式为TXT且已删除的Object恢复为最近的版本。

```
./ossutil64 revert-versioning oss://examplebucket --include "*.txt" -r
```

- 结合 `-t` 选项，将目标存储空间examplebucket下所有格式不为JPG且已删除的Object恢复为最近的版本。

```
./ossutil64 revert-versioning oss://examplebucket --exclude "*.jpg" -r
```

- 返回结果

恢复成功后，返回结果中将包含恢复的Object个数以及完成恢复操作所用时长，示例如下：

```
revert versioning object count is 10, batch list count is 10
0.066413(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过 `-e` 选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过 `-i` 选项指定该账号的AccessKey ID，并通过 `-k` 选项指定该账号的AccessKey Secret。

例如您需要将另一个阿里云账号下，华东2（上海）地域下目标存储空间testbucket内处于删除状态的exampletest.png文件恢复为最近的版本，命令如下：

```
./ossutil64 revert-versioning oss://testbucket/exampletest.png -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.34. rm（删除）

您可以通过 `rm` 命令删除不再需要的文件（Object）、碎片（Part）或存储空间（Bucket），以免产生不必要的存储费用。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的`./ossutil64`替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加 `--sign-version`、`--region`以及 `--cloudbox-id`选项。关于这三个选项的具体用法，请参见[通用选项](#)。

命令格式

```
./ossutil64 rm oss://bucketname[/prefix]
[-r, --recursive]
[-b, --bucket]
[-m, --multipart]
[-a, --all-type]
[-f, --force]
[--include <value>]
[--exclude <value>]
[--version-id <value>]
[--all-versions]
[--payer <value>]
[--encoding-type <value>]
```

参数及选项说明如下：

配置项	说明
bucketname	Bucket名称。
prefix	Bucket下的资源、例如目录、文件等。
-r, --recursive	如果指定该选项时，ossutil将删除Bucket下所有符合prefix条件的Object。如果不指定该选项，则ossutil只删除指定Object。
-b, --bucket	仅在删除Bucket时使用此选项。
-m, --multipart	指定操作的对象为Bucket中未完成的Multipart事件。
-a, --all-type	指定操作的对象为Bucket中符合prefix条件的Object和未完成的Multipart事件。
-f, --force	强制操作，不进行询问提示。
--include	包含符合指定条件的所有Object。
--exclude	不包含任何符合指定条件的Object。
--version-id	Object的指定版本。仅适用于已开启或暂停版本控制状态Bucket下的Object。
--all-versions	Object的所有版本。仅适用于已开启或暂停版本控制状态Bucket下的Object，且同一个删除示例中仅允许选择--version-id或--all-versions其中一个选项。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为requester。
--encoding-type	对 oss://bucket_name 之后的prefix进行编码，取值为url。如果不指定该选项，则表示prefix未经过编码。

删除Object

 **警告** Object删除后无法恢复，请谨慎操作。

- 删除示例

- 删除单个Object

删除examplebucket下名为exampleobject.txt文件。

```
./ossutil64 rm oss://examplebucket/exampleobject.txt
```

- 删除examplebucket下与前缀test匹配的所有文件。

```
./ossutil64 rm oss://examplebucket/test -r
```

- 删除examplebucket下后缀为.png的所有文件。

```
./ossutil64 rm oss://examplebucket --include "*.png" -r
```

- 删除examplebucket下文件名包含abc，且后缀不是.jpg和.txt的文件。

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

- 删除已开启版本控制的examplebucket下exampleobject.txt的指定版本。

```
./ossutil64 rm oss://examplebucket/exampleobject.txt --version-id CAEQARiBgID8rumR2hYiIGUyOTAyZGY2MzU5MjQ5ZjNhYzQzZjNlYTAYZDE3****
```

有关获取Object所有版本的具体操作，请参见[ls（列举账号级别下的资源）](#)。

- 删除已开启版本控制的examplebucket下exampleobject.txt的所有版本。

```
./ossutil64 rm oss://examplebucket/exampleobject.txt --all-versions
```

- 删除已开启版本控制的examplebucket下所有Object的所有版本。

```
./ossutil64 rm oss://examplebucket --all-versions -r
```

- 返回结果

以上示例删除成功后，返回结果中将包含删除的Object个数以及完成删除操作所用时长，示例如下：

```
Succeed: Total 8 objects. Removed 8 objects.  
0.106852(s) elapsed
```

删除Part

- 删除示例

- 结合-m选项删除examplebucket下exampleobject.txt中未完成的Multipart事件所产生的Part。

```
./ossutil64 rm -m oss://examplebucket/exampleobject.txt
```

- 结合-m和-r选项递归删除examplebucket下与指定前缀test匹配的所有文件中未完成的Multipart事件所产生的Part。

```
./ossutil64 rm -m oss://examplebucket/test -r
Do you really mean to remove recursively multipart uploadIds of oss://examplebucket/test(y or N)? y
```

- 结合-a和-r选项递归删除examplebucket下与指定前缀src匹配的所有已上传完成文件、以及未完成的Multipart事件所产生的Part。

```
./ossutil64 rm oss://examplebucket/src -a -r
Do you really mean to remove recursively objects and multipart uploadIds of oss://examplebucket/src(y or N)? y
```

● 返回结果

以上示例删除成功后，返回结果中将包含删除的Object数量、Part对应的UploadID数量以及完成删除操作所用时长，示例如下：

```
Succeed: Total 1 objects, 3 uploadIds. Removed 1 objects, 3 uploadIds.
1.922915(s) elapsed
```

删除Bucket

- 删除不包含Object或Part的examplebucket。

```
./ossutil64 rm oss://examplebucket -b
Do you really mean to remove the Bucket: examplebucket(y or N)? y
```

返回结果中包含删除的Bucket名称以及完成删除操作所用时长。

```
Removed Bucket: examplebucket
2.230745(s) elapsed
```

- 删除examplebucket及其包含的所有Object和Part。



警告 此操作将清除Bucket中的所有数据，且删除后不可恢复。请谨慎操作。

```
./ossutil64 rm oss://examplebucket -b -a -r
Do you really mean to remove recursively objects and multipart uploadIds of oss://examplebucket(y or N)? y
Do you really mean to remove the Bucket: examplebucket(y or N)? y
```

返回结果中包含删除的Object数量、Part对应的UploadID数量、Bucket名称完成删除操作所用时长。

```
Succeed: Total 189 objects, 37 uploadIds. Removed 189 objects, 37 uploadIds.
Removed Bucket: examplebucket
9.184193(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要删除另一个阿里云账号下，华东2（上海）地域下目标存储空间testbucket下的exampletest.png文件，命令如下：

```
./ossutil64 rm oss://testbucket/exampletest.png -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.35. set-acl（设置或修改ACL）

ACL是授予存储空间（Bucket）和文件（Object）访问权限的访问策略。您可以在创建Bucket或上传Object时设置ACL，也可以在创建Bucket或上传Object后的任意时间内修改ACL。**set-acl**命令用于设置或修改Bucket或Object的访问权限ACL。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 当您在OSS ON云盒中使用该命令时：
 - i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
 - ii. 在本文已有示例的基础上添加--sign-version、--region以及--cloudbox-id选项。关于这三个选项的具体用法，请参见[通用选项](#)。

设置或修改Bucket ACL

- 命令格式

```
./ossutil64 set-acl oss://bucketname acl -b [--retry-times <value>]
```

参数及选项说明如下：

配置项	说明
bucketname	待设置或修改ACL的Bucket名称。
acl	Bucket的读写权限ACL。取值如下： ◦ private（默认值）：只有该Bucket的拥有者可以对该Bucket内的文件进行读写操作，其他人无法访问该Bucket内的文件。 ◦ public-read：只有Bucket拥有者可以对该Bucket内的文件进行写操作，其他用户（包括匿名访问者）都可以对该Bucket中的文件进行读操作。这有可能造成您数据的外泄以及费用激增，若被人恶意写入违法信息还可能侵害您的合法权益。除特殊场景外，不建议您配置公共读写权限。 ◦ public-read-write：任何人（包括匿名访问者）都可以对该Bucket内文件进行读写操作。这有可能造成您数据的外泄以及费用激增，请谨慎操作。
-b	不添加此选项，默认对Object设置ACL。如果需要对Bucket设置ACL时，须添加此选项。
--retry-times	发生错误后的重试次数。默认值为10，取值范围为1~500。

● 使用示例

设置目标存储空间examplebucket的ACL为private。

```
./ossutil64 set-acl oss://examplebucket private -b
```

设置或修改Object ACL

● 命令格式

```
./ossutil64 set-acl oss://bucketname[/prefix]acl  
[-r]  
[--include <value>]  
[--exclude <value>]  
[--version-id <value>]  
[--job <value>]  
[--retry-times <value>]  
[--encoding-type <value>]
```

参数及选项说明如下：

配置项	说明
bucketname	Bucket名称。
prefix	Bucket下的资源，例如目录、文件等。
acl	Object的读写权限ACL。取值如下： ◦ <i>default</i>：继承Bucket的读写权限。 ◦ <i>private</i>（默认值）：只有该Bucket的拥有者可以对该Bucket内的文件进行读写操作，其他人无法访问该Bucket内的文件。 ◦ <i>public-read</i>：只有Bucket拥有者可以对该Bucket内的文件进行写操作，其他用户（包括匿名访问者）都可以对该Bucket中的文件进行读操作。这有可能造成您数据的外泄以及费用激增，若被人恶意写入违法信息还可能会侵害您的合法权益。除特殊场景外，不建议您配置公共读写权限。 ◦ <i>public-read-write</i>：任何人（包括匿名访问者）都可以对该Bucket内文件进行读写操作。这有可能造成您数据的外泄以及费用激增，请谨慎操作。
-r	如果指定该选项时，ossutil会对Bucket下所有符合prefix条件的Object设置ACL。如果不指定该选项，则ossutil只对cloud_url中指定的单个Object设置ACL。
--include	包含符合指定条件的所有Object。
--exclude	不包含任何符合指定条件的Object。
--version-id	Object的指定版本ID。仅适用于已开启或暂停版本控制状态Bucket下的Object。
--job	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。
--retry-times	发生错误后的重试次数。默认值为10，取值范围为1~500。

配置项	说明
--encoding-type	对 <code>oss://bucket_name</code> 之后的prefix进行编码，取值为 <code>url</code> 。如果不指定该选项，则表示prefix未经过编码。

● 使用示例

- 将目标存储空间examplebucket下的exampleobject.txt文件的读写权限ACL设置为private。

```
./ossutil64 set-acl oss://examplebucket/exampleobject.txt private
```

- 将目标存储空间examplebucket下指定版本ID为 `CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5Zj1hYzQzZjNlYTAyZDE3****` 的exampleobject.txt文件的读写权限ACL设置为private。

```
./ossutil64 set-acl oss://examplebucket/exampleobject.txt private --version-id CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5Zj1hYzQzZjNlYTAyZDE3****
```

- 将目标存储空间examplebucket下所有符合指定前缀为test的文件读写权限ACL设置为default。

```
./ossutil64 set-acl oss://examplebucket/test default -r
```

- 将目标存储空间examplebucket下所有后缀为.jpg的文件读写权限设置为private。

```
./ossutil64 set-acl oss://examplebucket private --include "*.jpg" -r
```

- 将目标存储空间examplebucket下文件名称包含abc，且后缀不是.png和.txt的文件读写权限设置为default。

```
./ossutil64 set-acl oss://examplebucket default --include "**abc*" --exclude "*.png" --exclude "*.txt" -r
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要获取另一个阿里云账号下，华东2（上海）地域下名为testbucket的存储空间设置读写权限ACL为private，命令如下：

```
./ossutil64 set-acl oss://testbucket private -b -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.36. set-meta（管理文件元信息）

文件元信息是对文件（Object）的属性描述，包括HTTP标准属性（HTTP Header）和用户自定义元数据（User Meta）两种。其中，HTTP Header可用于自定义HTTP请求的策略，用户自定义元数据可用于标识文件的用途或属性等。您可以通过set-meta命令为已上传的文件（Object）设置、修改或者删除文件元信息。

说明

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 通过set-meta命令设置或更新文件元信息后，您可以通过stat命令查看文件元信息。更多信息，请参见[stat](#)。

命令格式

```
./ossutil64 set-meta oss://bucketname[/prefix]
[header:value#header:value...]
[--update]
[--delete]
[-r, --recursive]
[-f, --force]
[--object-file, <value>]
[--snapshot-path, <value>]
[--disable-ignore-error]
[--version <value>]
[--include <value>]
[--exclude <value>]
```

参数及选项说明如下：

配置项	说明
bucketname	Bucket名称。
prefix	Bucket下的资源，例如目录、文件等。
header:value#header:value...	以键值对的形式标识元信息。其中，Header不区分大小写，但Value区分大小写。如果需要设置多组元信息，则各组元信息之间以井号（#）连接。例如 Cache-Control:no-cache#Expires:2022-10-12T00:00:00.000Z 。 当前可选的Header列表如下： 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 以及以X-Oss-Meta-开头的header 关于Header的用法，请参见管理文件元信息。

配置项	说明
--update	更新文件元信息。该选项可简写为 <i>-u</i> ，且不能与 <i>--delete</i> 选项同时使用。
--delete	删除文件元信息。该选项不能与 <i>--update</i> 选项同时使用。
-r, --recursive	如果指定该选项时，ossutil将为所有与prefix匹配的文件设置元信息。如果不指定该选项，则ossutil只为指定文件设置元信息。
-f, --force	强制操作，不进行询问提示。
--object-file	该选项用于批量设置、更新或删除多个Object的元信息。使用该选项的步骤如下： - 您需要指定.txt或者XML格式的本地文件，并在本地文件中换行填写所有待设置、更新或删除元信息的Object。 - ossutil读取本地文件中的所有Object，然后对这些Object进行元信息的设置、更新或删除操作。  说明 当其中一个Object操作出现错误时，ossutil会将该Object的错误信息记录到report文件，并继续设置其他Object的元信息。成功操作的Object信息不会被记录到report文件中。
--snapshot-path	如果指定该选项，则只对本次操作的Object生成快照。如果操作的Object已经存在快照信息，则忽略本次操作。  说明 该选项需结合 <i>-r</i>、<i>--recursive</i> 或 <i>--object-file</i> 选项使用。
--disable-ignore-error	批量操作时不忽略错误。
--version-id	Object的指定版本。仅适用于已开启或暂停版本控制状态Bucket下的Object。
--include	包含符合指定条件的所有Object。
--exclude	不包含任何符合指定条件的Object。

设置或更新元信息

在以下未包含 *--update* 选项的使用示例中，如果未指定 *header:value#header:value...* 选项，则仅保留以 *X-Oss-Meta-* 开头的自定义元信息，且不更改其Value值。对于HTTP Header信息，ossutil将进入交互模式并要求您确认是否继续保留。

- 将examplebucket中exampleobject.txt的读写权限设置为私有（private）。

```
./ossutil64 set-meta oss://examplebucket/exampleobject.txt X-Oss-Object-Acl:private
```

- 将examplebucket中exampleobject.txt的指定版本存储类型（X-Oss-Storage-Class）设置为标准存储（Standard）。

```
./ossutil64 set-meta oss://examplebucket/exampleobject.txt X-Oss-Storage-Class:Standard
--version-id CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5ZjlhYzQzZjNlYTAyZDE3MDRk
```

有关获取Object版本的具体操作，请参见[ls](#)。

- 将examplebucket中exampleobject.txt的读写权限ACL更新为公共读（public-read）。

```
./ossutil64 set-meta oss://examplebucket/exampleobject.txt X-Oss-Object-Acl:public-read
--update
```

使用--update选项时仅更新目标Object中与指定Header匹配的元信息，并替换为该Header对应的Value值，其中Value值可以为空。Value置空时，表示保留原有的Value信息。例如，该示例表示仅将名为exampleobject.txt的Object的读写权限更新为public-read，Object的其他元信息保留不变。

示例一：为单个Object设置或更新元信息

- 为多个Object同时设置或修改元信息时，如果其中一个Object操作出现错误，ossutil会将该Object的错误信息记录到ossutil_output目录下的report文件中。成功操作的Object信息不会记录到该report文件。
- 为与指定前缀匹配的Object设置元信息

结合-r选项，将examplebucket中与指定前缀src匹配的所有Object的缓存行为（Cache-Control）修改为no-cache（不直接使用缓存），读写权限（x-oss-object-acl）修改为private（私有）。

```
./ossutil64 set-meta oss://examplebucket/src Cache-Control:no-cache#X-Oss-Object-Acl:private -r
```

- 为与指定过滤条件匹配的Object修改元信息
 - 结合-r及-l选项，将examplebucket中目录desfolder下所有文件类型为.jpg的Object的存储类型修改为低频访问类型（IA）。

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

- 结合-r及-l选项，将examplebucket中目录desfolder下文件名包含abc且文件类型不为.jpg和.txt的Object的存储类型修改为归档存储（Archive）。

```
./ossutil64 set-meta oss://examplebucket/desfolder/ X-Oss-Storage-Class:Archive --inc
lude "abc*" --exclude "*.jpg" --exclude "*.txt" -u -r
```



注意 通过set-meta命令对大于100 MB的冷归档存储类型Object进行存储类型转换时，可能会出现因服务器超时导致转换操作失败的问题。因此，当您需要对大于100 MB的冷归档存储类型Object进行存储类型转换时，请使用cp命令。

示例二：为相同目录下的多个Object批量设置或更新元信息

- 不同目录下

当您需要为Bucket中的多个Object设置或更新元信息，这些Object归属于Bucket的不同目录下，例如exampleobject1.jpg位于examplebucket根目录下，exampleobject2.png位于examplebucket的dir1/目录下，exampleobject3.txt位于examplebucket的dir2/目录下。此时，您需要按照以下步骤设置或更新多个Object的元信息。

- a. 将待设置或更新元信息的多个Object写入本地文件localfile.txt。

```
exampleobject1.jpg
dir1/exampleobject2.png
dir2/exampleobject3.txt
```

- b. 为多个指定Object设置或更新元信息。

结合--object-file选项将examplebucket中多个Object的缓存行为（Cache-Control）修改为no-cache（不直接使用缓存），读写权限（x-oss-object-acl）修改为private（私有），并开启快照。

```
./ossutil64 set-meta oss://examplebucket Cache-Control:no-cache#X-Oss-Object-Acl:private --update --object-file localfile.txt --snapshot-path dir/
```

- 相同目录下

- 方法一

参考[不同目录下的操作示例](#)为相同目录下的多个Object设置或更新元信息。

- 方法二：

参考[示例二](#)为相同目录下的多个Object设置或更新元信息。

示例三：为相同或不同目录下的多个Object设置或更新元信息

- 以上操作成功后，返回结果中将包含设置或更新文件元信息所用时长，示例如下：

```
0.106852(s) elapsed
```

返回结果

删除自定义元信息

 **注意** 结合--delete选项仅支持删除以 X-Oss-Meta- 开头的自定义元信息。

- 删除examplebucket中exampleobject.txt包含的以 X-Oss-Meta- 开头的自定义元信息，其中Value值必须为空。

```
./ossutil64 set-meta oss://examplebucket/exampleobject.txt X-Oss-Meta-Createdby --delete
```

示例一：删除单个Object的自定义元信息

- 批量删除多个Object的自定义元信息步骤如下：

- i. 将待删除自定义元信息的Object名称写入本地文件localfile.txt。

```
exampleobject1.jpg
dir1/exampleobject2.png
dir2/exampleobject3.txt
```

ii. 删除多个Object的自定义元信息。

结合--object-file选项批量删除examplebucket中多个Object包含的以 X-Oss-Meta- 开头的自定义元信息，其中Value值必须为空。

```
./ossutil64 set-meta oss://examplebucket X-Oss-Meta-Location --delete --object-file localfile.txt --snapshot-path dir/
```

示例二：批量删除多个Object的自定义元信息

- 以上操作成功后，返回结果中将包含删除自定义元信息所用时长，示例如下：

```
0.106846(s) elapsed
```

返回结果

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东2（上海）地域下目标存储空间testbucket下的testobject.jpg文件设置读写权限ACL为private的元信息，命令如下：

```
./ossutil64 set-meta oss://testbucket/testobject.jpg X-Oss-Object-Acl:private -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.37. sign（生成签名URL）

当文件（Object）上传至存储空间（Bucket）后，您可以通过sign生成经过签名的文件URL，并将签名URL分享给第三方供其下载或预览。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

```
./ossutil64 sign cloud_url
[--timeout <value>]
[--version-id <value>]
[--traffic-limit <value>]
[--disable-encode-slash]
[--payer <value>]
```

参数及选项说明如下：

配置项	说明
cloud_url	文件所在Bucket的完整路径。

配置项	说明
--timeout	签名URL过期时间，单位为秒。默认值为60。  注意 当前时间戳与签名URL的过期时间之和不能超过9223372036854775807，否则会溢出报错。例如，当前时间戳为1643341269，则签名URL过期时间最大不能超过9223372035211434538。
--version-id	Object的指定版本ID。仅适用于已开启或暂停版本控制状态Bucket下的Object。
--traffic-limit	限定HTTP的访问速度，单位为bit/s。缺省值为0，表示不受限制。取值范围为819200~838860800，即100 KB/s~100 MB/s。
--disable-encode-slash	不对cloud_url中携带的正斜线（/）进行编码。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为requester。

使用示例

- 为目标存储空间examplebucket下的文件exampleobject.png生成文件URL，其默认超时时间为60秒。

```
./ossutil64 sign oss://examplebucket/exampleobject.png
```
- 为目标存储空间examplebucket下的文件exampleobject.png生成文件URL，并指定超时时间为3600秒。

```
./ossutil64 sign oss://examplebucket/exampleobject.png --timeout 3600
```
- 为目标存储空间examplebucket下的文件exampleobject.png生成文件URL，指定其超时时间为7200秒，并限定其访问速度为100MB/s。

```
./ossutil64 sign oss://examplebucket/exampleobject.png --timeout 7200 --traffic-limit 838860800
```
- 为已开启版本控制的存储空间examplebucket下的exampleobject.jpg的指定版本文件生成文件URL，并指定超时时间为1800秒。

```
./ossutil64 sign oss://examplebucket/exampleobject.jpg --timeout 1800 --version-id CAEQARiBgID8rumR2hYiIGUyOTAYZGY2MzU5MjQ5Zj1hYzQzZjNlYTAYZDE3****
```
- 以上示例操作成功后，返回结果中将包含生成文件URL所用时长、文件URL过期时间以及签名等信息，示例如下：

```
https://examplebucket.ss-cn-hangzhou.aliyuncs.com/exampleobject.png?Expires=1608282224&OSSAccessKeyId=LTAI4G33piUmgRN1DXx9****&Signature=jo4%2FGykfuclA4fvyvKRpRyymYH****
0.368676(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东2（上海）地域下存储空间testbucket下的exampletest.jpg文件生成超时时间为3600秒的文件URL，命令如下：

```
./ossutil64 sign oss://testbucket/exampletest.jpg --timeout 3600 -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.38. stat（查看Bucket和Object信息）

stat命令用于查看指定存储空间（Bucket）或者对象（Object）的相关信息。例如，您可以通过该命令查看Bucket的存储类型、Object元信息等。

 注意

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 只有Bucket拥有者及授予了 `oss:GetObject`、`oss:GetObjectAcl` 和 `oss:GetBucketInfo` 权限的RAM用户允许使用此命令查看Object元信息。

命令格式

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

参数及选项说明如下：

配置项	说明
bucketname	目标Bucket名称。
objectname	目标Object名称。
--encoding-type	对Object名称进行编码，取值为url。如果不指定该选项，则表示Object名称未经过编码。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为requester。
--version-id	Object的指定版本。仅适用于已开启或暂停版本控制状态Bucket下的Object。

使用示例

- 查看目标存储空间examplebucket的相关信息。

```
./ossutil64 stat oss://examplebucket
```

以下返回结果表明已成功获取examplebucket的相关信息，包括Bucket名称、所在地域、创建时间、读写权限ACL等。

```
Name           : examplebucket
Location        : oss-cn-hangzhou
CreationDate    : 2021-06-30 16:04:41 +0800 CST
ExtranetEndpoint : oss-cn-hangzhou.aliyuncs.com
IntranetEndpoint : oss-cn-hangzhou-internal.aliyuncs.com
ACL             : private
Owner           : 148562088256****
StorageClass    : Standard
RedundancyType  : LRS
0.132413(s) elapsed
```

- 查看目标存储空间examplebucket下名为exampleobject.jpg的相关信息。

```
./ossutil64 stat oss:///examplebucket/exampleobject.jpg
```

以下返回结果表明已成功获取exampleobject.jpg的相关信息，包括读写权限ACL、Content-Type等文件元信息、Etag值、最后更新时间等。

```
ACL             : default
Accept-Ranges   : bytes
Content-Length  : 8746
Content-Md5     : cvc5wcklut76CVoEi2UW****
Content-Type    : image/jpeg
Etag            : 72F739C1C925BADEFA095A048B65****
Last-Modified   : 2021-07-27 09:53:03 +0800 CST
Owner           : 148562088256****
X-Oss-Hash-Crc64ecma : 56037670485008****
X-Oss-Object-Type : Normal
X-Oss-Storage-Class : Standard
0.194739(s) elapsed
```

- 查看目标存储空间examplebucket下名称中含特殊字符的Object信息

例如，您需要查看examplebucket下名为示例.txt的Object信息。由于ossutil目前只支持以URL编码的方式输出或输入Object名称，对于无法输入或识别的特殊字符，需要对特殊字符进行URL编码之后再输出。

```
./ossutil64 stat oss:///examplebucket/%E7%A4%BA%E4%BE%8B.txt --encoding-type url
```

以下返回结果表明已成功获取 示例.txt 的相关信息，包括读写权限ACL、Content-Type等文件元信息、Etag值、最后更新时间等。

```

ACL                : default
Accept-Ranges      : bytes
Content-Length     : 113089
Content-Md5        : mAYgHM5cSkZNBxKDDqgW****
Content-Type       : image/jpeg
Etag               : 9806201CCE5C4A464D0712830EAA****
Last-Modified      : 2021-07-27 10:08:57 +0800 CST
Owner              : 148562088256****
X-Oss-Hash-Crc64ecma : 1106061527435521****
X-Oss-Object-Type  : Normal
X-Oss-Storage-Class : Standard
0.168335(s) elapsed

```

- 查看目标存储空间examplebucket下指定版本Object的相关信息

```

./ossutil64 stat oss://examplebucket/test.jpg --version-id CAEQFRiBgICw.YSX1xciIDNiNzc3ZTdmNTQzOTQ1OTM4MWUwOWI1Y2MlZTgz****

```

有关获取Object版本的具体操作，请参见[ls](#)。

以下返回结果表明已成功获取指定版本Object的相关信息，包括读写权限ACL、Content-Type等文件元信息、Etag值、最后更新时间、版本ID等。

```

ACL                : default
Accept-Ranges      : bytes
Content-Length     : 190540
Content-Md5        : aJQnxHQ2b+zW6hh0c+2y****
Content-Type       : image/jpeg
Etag               : 689427C474366FECD6EA187473ED****
Last-Modified      : 2021-07-27 10:19:25 +0800 CST
Owner              : 148562088256****
X-Oss-Hash-Crc64ecma : 1666171672179499****
X-Oss-Object-Type  : Normal
X-Oss-Storage-Class : Standard
X-Oss-Version-Id   : CAEQFRiBgICw.YSX1xciIDNiNzc3ZTdmNTQzOTQ1OTM4MWUwOWI1Y2MlZTgz****
0.213528(s) elapsed

```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要获取另一个阿里云账号下，华东2（上海）地域下名为testbucket的存储空间信息，命令如下：

```

./ossutil64 stat oss://testbucket -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA*
*** -k 67DLVBkh7EamOjy2W5RVAHUY9H****

```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.39. sync（同步文件）

2.3.39.1. 简介

sync命令用于同步本地文件到OSS、同步OSS文件到本地、或者在OSS之间同步文件。

有关sync命令用于同步本地文件到OSS的命令格式及使用示例的更多信息，请参见[同步本地文件到OSS](#)。

有关sync命令用于同步OSS文件到本地的命令格式及使用示例的更多信息，请参见[同步OSS文件到本地](#)。

有关sync命令用于在OSS之间同步文件的命令格式及使用示例的更多信息，请参见[在OSS之间同步文件](#)。

2.3.39.2. 同步本地文件到OSS

sync命令用于将本地文件同步到OSS。

注意事项

- Binary名称

本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

- 文件数量

通过sync命令执行同步任务时，如果没有携带--delete选项，则单次同步任务同步的文件个数无限制。如果携带了--delete选项，则单次同步任务最多可同步100万个文件。当同步的文件个数超出100万时，将报错 `over max sync numbers 1000000.`。

- sync命令与cp命令的区别

- sync命令强制以递归的方式遍历指定文件夹内所有文件或子文件夹。cp命令需增加-r选项才会进行递归操作。
- 通过sync命令将数据同步到OSS时，ossutil支持通过--delete选项将目的端存在而源端不存在的文件都删除，仅保留本次同步的文件。cp命令不支持--delete选项。
- sync不支持--version-id选项，无法在已开启版本控制的Bucket内同步历史版本文件。cp命令支持--version-id选项。

除以上区别外，sync命令与cp命令用法类似。有关cp命令的用法及示例，请参见[cp（上传文件）](#)。

命令格式

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

参数及选项说明如下：

配置项	说明
file_url	待同步的本地文件夹路径。例如Linux系统文件路径 <code>/localfolder/</code> ，Windows系统文件路径 <code>D:\localfolder\</code> 。
cloud_url	OSS文件夹路径。格式为 <code>oss://bucketname/path/</code> 。例如 <code>oss://examplebucket/exampledir/</code> 。如果输入的 <code>cloud_url</code> 没有以正斜线（/）结尾，ossutil会自动在结尾处添加一个正斜线（/）。
-f --force	强制操作，不进行询问提示。
-u, --update	只有当目标文件不存在，或源文件的最后修改时间晚于目标文件时，ossutil才会执行同步操作。
--delete	删除目的端指定路径下的其他文件，仅保留本次同步的文件。  警告 建议您使用--delete选项前开启版本控制，防止数据误删除。
--enable-symlink-dir	同步链接子目录。
--disable-all-symlink	同步目录时，忽略所有的链接子文件以及链接子目录。

配置项	说明
--disable-ignore-error	批量操作时不忽略错误。
--only-current-dir	仅同步当前目录下的文件，忽略子目录及子目录下的文件。
--output-dir	指定输出文件所在的目录。输出文件是指批量同步文件出错时产生的report文件，默认保存在当前目录下的ossutil_output目录。
-bigfile-threshold	设置断点续传文件的大小阈值，单位为字节。 默认值：100 MB 取值范围：0~9223372036854775807
--part-size	设置分片大小，单位为字节。默认情况下ossutil会根据文件大小自行计算合适的分片大小值。 取值范围：1~9223372036854775807
--checkpoint-dir	指定断点续传记录信息所在的目录。断点续传操作失败时，ossutil会自动创建名为 .ossutil_checkpoint 的目录，并在该目录下记录checkpoint信息，断点续传成功后会删除该目录。如果通过该选项指定了目录，请确保指定的目录可以被删除。
--encoding-type	文件名称的编码方式。取值为url。如果不指定该选项，则表示文件名称未经过编码。
--snapshot-path	指定保存同步文件时的快照信息所在的目录。在下次同步文件时，ossutil会读取指定目录下的快照信息进行增量同步。
--include	包含符合指定条件的所有文件。
--exclude	不包含任何符合指定条件的文件。
--meta	设置文件的元信息，格式为 header:value#header:value ，示例为 Cache-Control:no-cache#Content-Encoding:gzip 。有关元信息的介绍，请参见 set-meta（管理文件元信息） 。
--acl	文件的读写权限ACL。取值如下： • default：继承Bucket的读写权限。 • private（默认值）：只有该Bucket的拥有者可以对该Bucket内的文件进行读写操作，其他人无法访问该Bucket内的文件。 • public-read：只有Bucket拥有者可以对该Bucket内的文件进行写操作，其他用户（包括匿名访问者）都可以对该Bucket中的文件进行读操作。这有可能造成您数据的外泄以及费用激增，若被人恶意写入违法信息还可能会侵害您的合法权益。除特殊场景外，不建议您配置此权限。 • public-read-write：任何人（包括匿名访问者）都可以对该Bucket内文件进行读写操作。这有可能造成您数据的外泄以及费用激增，请谨慎操作。
--maxupspeed	最大上传速度，单位为KB/s。默认值为0，表示不限制上传速度。
--disable-crc64	关闭CRC64数据校验。

配置项	说明
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <i>requester</i> 。
-j, --job	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。
--parallel	单文件操作时的并发任务数，取值范围为1~10000。如果不设置此选项，默认由ossutil根据操作类型和文件大小自行决定。
--retry-times	发生错误后的重试次数。默认值为10，取值范围为1~500。
--tagging	文件的标签信息，格式为 <code>TagkeyA=TagvalueA&TagkeyB=TagvalueB....</code> 。

使用示例

本地根目录下 *localfolder* 文件夹内有 *d.txt* 和 *e.png* 文件。OSS 中名为 *examplebucket* 的 Bucket 下文件夹 *destfolder* 内有文件 *a.txt*、*b.txt* 和子文件夹 *C*，文件结构如下：

本地根目录	examplebucket
└─ localfolder	└─ destfolder/
├─ d.txt	├─ a.txt
└─ e.png	├─ b.txt
	└─ C/

示例场景及命令如下：

- 将本地 *localfolder* 文件夹同步到 OSS

```
./ossutil64 sync localfolder/ oss://examplebucket/destfolder/
```

命令执行后，*examplebucket* 的 *destfolder* 文件夹内新增 *d.txt* 和 *e.png* 文件。

本地根目录	examplebucket
└─ localfolder	└─ destfolder/
├─ d.txt	├─ a.txt
└─ e.png	├─ b.txt
	├─ d.txt
	├─ e.png
	└─ C/

- 同步本地文件夹到 OSS，并删除 OSS 指定路径下已存在而本地端不存在的文件

通过增加 *--delete* 选项，删除目的端存在而源端不存在的文件，仅保留本次同步的文件。

```
./ossutil64 sync localfolder/ oss://examplebucket/destfolder/ --delete
```

命令执行后，本地 *localfolder* 文件夹被同步到 OSS，同时会删除 OSS 的文件 *a.txt*、*b.txt* 以及子文件夹 *C*，即 OSS 的 *destfolder* 文件夹内仅保留 *d.txt* 和 *e.png* 文件。

```

本地根目录          examplebucket
└─ localfolder      └─ destfolder/
    │               │
    └─ d.txt         └─ d.txt
    └─ e.png         └─ e.png
  
```

- 同步本地文件夹到OSS，并省略询问操作

默认情况下，同步本地文件夹到OSS时，如果目的端存在同名文件，ossutil会进行覆写操作的询问。命令如下：

```

./ossutil64 sync localfolder/ oss://examplebucket/destfolder/
cp: overwrite "oss://examplebucket/destfolder/d.txt"(y or N)?
  
```

如果您确认目的端文件均可被覆写，可增加 `-f`、`--force` 选项强制执行覆写操作。命令如下：

```

./ossutil64 sync localfolder/ oss://examplebucket/destfolder/ -f, --force
  
```

命令执行后，examplebucket的destfolder文件夹内新增d.txt和e.png文件。

```

本地根目录          examplebucket
└─ localfolder      └─ destfolder/
    │               │
    └─ d.txt         └─ a.txt
    └─ e.png         └─ b.txt
                     └─ d.txt
                     └─ e.png
                     └─ C/
  
```

- 以上示例同步成功后，返回结果中将包含同步的文件数量、文件大小以及完成同步操作所用时长，示例如下：

```

Succeed: Total num: 2, size: 750,081. OK num: 2(upload 2 files).
average speed 1641000(byte/s)
  
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过 `-e` 选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过 `-i` 选项指定该账号的AccessKey ID，并通过 `-k` 选项指定该账号的AccessKey Secret。

例如，您需要将本地文件夹srcfolder的文件同步至另一个阿里云账号下，华东2（上海）地域下存储空间examplebucket的文件夹testfolder，命令如下：

```

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.39.3. 同步OSS文件到本地

`sync` 命令用于将OSS中的文件同步到本地。

注意事项

- Binary名称

本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

- 文件数量
通过sync命令执行同步任务时，如果没有携带--delete选项，则单次同步任务同步的文件个数无限制。如果携带了--delete选项，则单次同步任务最多可同步100万个文件。当同步的文件个数超出100万时，将报错 `over max sync numbers 1000000.`。
- sync命令与cp命令的区别
 - sync命令强制以递归的方式遍历指定文件夹内所有文件或子文件夹。cp命令需增加-r选项才会进行递归操作。
 - 通过sync命令将数据同步到OSS时，ossutil支持通过--backup-dir选项指定目标文件夹，用于保存目的端存在而源端不存在的文件。cp命令不支持--backup-dir选项。
 - sync不支持--version-id选项，无法在已开启版本控制的Bucket内同步历史版本文件。cp命令支持--version-id选项。

除以上区别外，sync命令与cp命令用法类似。有关cp命令的用法及示例，请参见[cp（下载文件）](#)。

命令格式

```
./ossutil64 sync cloud_url file_url
[-f --force]
[-u --update]
[--maxdownspeed <value>]
[--delete]
[--backup-dir <value>]
[--enable-symlink-dir]
[--disable-all-symlink]
[--disable-ignore-error]
[--only-current-dir]
[--output-dir <value>]
[--bigfile-threshold <value>]
[--part-size <value>]
[--checkpoint-dir <value>]
[--range <value>]
[--encoding-type <value>]
[--snapshot-path <value>]
[--include <value>]
[--exclude <value>]
[--disable-crc64]
[--payer <value>]
[-j, --job <value>]
[--parallel <value>]
[--retry-times <value>]
```

参数及选项说明如下：

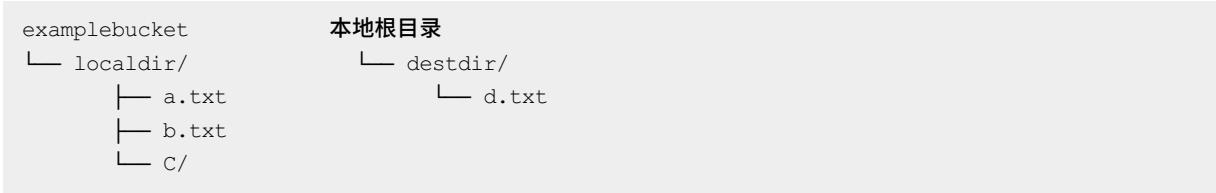
配置项	说明
-----	----

配置项	说明
cloud_url	OSS文件夹（目录）路径。格式为 <code>oss://bucketname/path/</code> 。例如 <code>oss://examplebucket/exampledir/</code> 。如果输入的 <code>cloud_url</code> 没有以正斜线（/）结尾，ossutil会自动在结尾处添加一个正斜线（/）。
file_url	待同步的本地文件夹路径。例如Linux系统文件夹路径 <code>/localfolder/</code> ，Windows系统文件夹路径 <code>D:\localfolder\</code> 。
-f --force	强制操作，不进行询问提示。
-u, --update	只有当目标文件不存在，或源文件的最后修改时间晚于目标文件时，ossutil才会执行同步操作。
--maxdownspeed	最大下载速度，单位为KB/s，默认值为0，表示不限制下载速度。
--delete	将目的端指定路径下的其他文件都删除，仅保留本次同步的文件。  警告 建议您使用--delete选项前开启 版本控制 ，防止数据被误删。
--backup-dir	指定目标文件夹，用于保存目的端存在而源端不存在的文件。
--enable-symlink-dir	同步链接子目录。
--disable-all-symlink	同步目录时，忽略所有的链接子文件以及链接子目录。
--disable-ignore-error	批量操作时不忽略错误。
--only-current-dir	仅同步当前目录下的文件，忽略子目录及子目录下的文件。
--output-dir	指定输出文件所在的目录。输出文件是指批量同步文件出错时产生的report文件，默认保存在当前目录下的 <code>ossutil_output</code> 目录。
-bigfile-threshold	设置断点续传文件的大小阈值，单位为字节。 默认值：100 MB 取值范围：0~9223372036854775807
--part-size	设置分片大小，单位为字节。默认情况下ossutil会根据文件大小自行计算合适的分片大小值。 取值范围：1~9223372036854775807
--checkpoint-dir	指定断点续传记录信息所在的目录。断点续传操作失败时，ossutil会自动创建名为 <code>.ossutil_checkpoint</code> 的目录，并在该目录下记录checkpoint信息，断点续传成功后会删除该目录。如果指定了该选项，请确保指定的目录可以被删除。

配置项	说明
--range	下载目标文件的指定字段，并保存为一个新的文件，字段从0开始编号。 - 指定区间 例如指定为 3-9 ，表示下载文件的第3个字节到第9个字节（包含第3和第9字节）。 - 指定开始位置 例如指定为 3- ，表示从第3个字节开始到文件结尾（包含第3个字节）。 - 指定结束位置 例如指定为 -9 ，表示从0字节到第9个字节（包含第9个字节）。
--encoding-type	文件名称的编码方式。取值为 url。如果不指定该选项，则表示文件名称未经过编码。
--snapshot-path	指定保存同步文件时的快照信息所在的目录。在下次同步文件时，ossutil 会读取指定目录下的快照信息进行增量同步。
--include	包含符合指定条件的所有文件。
--exclude	不包含任何符合指定条件的文件。
--disable-crc64	关闭CRC64数据校验。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 requester。
-j, --job	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。
--parallel	单文件操作时的并发任务数，取值范围为1-10000。如果不设置此选项，默认由ossutil根据操作类型和文件大小自行决定。
--retry-times	发生错误后的重试次数。默认值为10，取值范围为1~500。

使用示例

OSS存储空间examplebucket下localdir文件夹内有文件a.txt、b.txt以及子文件夹C，本地根目录下destdir文件夹内有d.txt文件。文件结构如下：



- 将OSS的localdir文件夹同步到本地

```
./ossutil64 sync oss://examplebucket/localdir/ destdir/
```

命令执行完毕后，本地destdir文件夹内新增a.txt、b.txt以及子文件夹C。


```

examplebucket          本地根目录
└─ localdir/           └─ destdir/
    │                   │
    ├── a.txt           ├── a.txt
    ├── b.txt           ├── b.txt
    └─ C/               └─ d.txt
                        └─ C/

```

- 将OSS存储空间examplebucket下的文件夹localdir同步到本地文件夹destdir，并通过--backup-dir选项指定backup文件夹，用于保存本地指定目录下已存在而OSS不存在的文件，同时在原有文件夹下的这些文件会被删除。

```
./ossutil64 sync oss://examplebucket/localdir/ destdir/ --delete --backup-dir backup/
```

命令执行完毕后，OSS的localdir文件夹将同步到本地，同时本地destdir文件夹内存在的而OSS不存在的文件被转移到backup文件夹内。即本地destdir文件夹仅保存a.txt、b.txt及子文件夹C，原有的d.txt文件将保存至backup文件夹，且d.txt文件在原有的destdir文件夹下被删除。

```

examplebucket          本地根目录
└─ localdir/           └─ destdir/
    │                   │
    ├── a.txt           ├── a.txt
    ├── b.txt           ├── b.txt
    └─ C/               ├── C/
                        └─ backup/
                            └─ d.txt

```

- 以上示例同步成功后，返回结果中将包含同步的文件数量、文件大小以及完成同步操作所用时长，示例如下：

```
Succeed: Total num: 2, size: 750,081. OK num: 2(upload 2 files).
average speed 1641000 (byte/s)
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要将另一个阿里云账号下，华东2（上海）地域下存储空间examplebucket的文件夹srcfolder同步至本地文件夹examplefolder，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.39.4. 在OSS之间同步文件

您可以使用sync命令将同一个地域下某个存储空间（Bucket）中某个文件夹下的文件（Object）批量同步到另一个Bucket。

注意事项

- Binary名称

本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

- 文件数量

通过sync命令执行同步任务时，如果没有携带--delete选项，则单次同步任务同步的文件个数无限制。如果携带了--delete选项，则单次同步任务最多可同步100万个文件。当同步的文件个数超出100万时，将报错 `over max sync numbers 1000000.`。

- sync命令与cp命令的区别

- sync命令强制以递归的方式遍历指定文件夹内所有文件或子文件夹。cp命令需增加-r选项才会进行递归操作。
- 通过sync命令将数据同步到OSS时，ossutil支持通过--delete选项将目标Bucket存在而源Bucket不存在的文件都删除，仅保留本次同步的文件。cp命令不支持--delete选项。
- sync不支持--version-id选项，无法在已开启版本控制的Bucket内同步历史版本文件。cp命令支持--version-id选项。

除以上区别外，sync命令与cp命令用法类似。关于cp命令的用法及示例，请参见[cp（拷贝文件）](#)。

使用限制

不支持将源Bucket中冷归档存储类型Object同步到目标Bucket。

命令格式

```
./ossutil64 sync cloud_url cloud_url
[-f --force]
[-u --update]
[--delete]
[--enable-symlink-dir]
[--disable-all-symlink]
[--disable-ignore-error]
[--only-current-dir]
[--output-dir <value>]
[--bigfile-threshold <value>]
[--part-size <value>]
[--checkpoint-dir <value>]
[--encoding-type <value>]
[--include <value>]
[--exclude <value>]
[--meta <value>]
[--acl <value>]
[--maxupspeed <value>]
[--disable-crc64]
[--payer <value>]
[-j, --job <value>]
[--parallel <value>]
[--retry-times <value>]
[--tagging <value>]
```

参数及选项说明如下：

配置项	说明
cloud_url	分别填写OSS源文件夹路径以及目标文件夹路径。格式为 <code>oss://bucketname/path/</code>。例如，将目标存储空间examplebucket的srcdir文件夹同步到destdir文件夹，则源文件夹路径填写为 <code>oss://examplebucket/srcdir/</code>，目标文件夹路径填写为 <code>oss://examplebucket/destcdir/</code>。  注意 如果输入的 <code>cloud_url</code> 没有以正斜线 (/) 结尾，ossutil会自动在结尾处添加一个正斜线 (/)。
-f --force	强制操作，不进行询问提示。
-u, --update	只有当目标文件不存在，或源文件的最后修改时间晚于目标文件时，ossutil才会执行同步操作。
--delete	将目标Bucket指定路径下的其他文件都删除，仅保留本次同步的文件。  警告 建议您使用--delete选项前开启版本控制，防止数据被误删。
--enable-symlink-dir	同步链接子目录。
--disable-all-symlink	同步目录时，忽略所有的链接子文件以及链接子目录。
--disable-ignore-error	批量操作时不忽略错误。
--only-current-dir	仅同步当前目录下的文件，忽略子目录及子目录下的文件。
--output-dir	指定输出文件所在的目录。输出文件是指批量同步文件出错时产生的report文件，默认保存在当前目录下的ossutil_output目录。
-bigfile-threshold	设置断点续传文件的大小阈值，单位为字节。 默认值：100 MB 取值范围：0~9223372036854775807
--part-size	设置分片大小，单位为字节。默认情况下ossutil会根据文件大小自行计算合适的分片大小值。 取值范围：1~9223372036854775807
--checkpoint-dir	指定断点续传记录信息所在的目录。断点续传操作失败时，ossutil会自动创建名为 <code>.ossutil_checkpoint</code> 的目录，并在该目录下记录checkpoint信息，断点续传成功后会删除该目录。如果指定了该选项，请确保指定的目录可以被删除。
--encoding-type	文件名称的编码方式。取值为url。如果不指定该选项，则表示文件名称未经过编码。

配置项	说明
--include	包含符合指定条件的所有文件。
--exclude	不包含任何符合指定条件的文件。
--meta	设置文件的元信息，格式为 <code>header:value#header:value</code> ，示例为 <code>Cache-Control:no-cache#Content-Encoding:gzip</code> 。有关元信息的介绍，请参见 set-meta（管理文件元信息） 。
--acl	文件的读写权限ACL。取值如下： • <i>default</i>：继承Bucket的读写权限。 • <i>private</i>（默认值）：只有该Bucket的拥有者可以对该Bucket内的文件进行读写操作，其他人无法访问该Bucket内的文件。 • <i>public-read</i>：只有Bucket拥有者可以对该Bucket内的文件进行写操作，其他用户（包括匿名访问者）都可以对该Bucket中的文件进行读操作。这有可能造成您数据的外泄以及费用激增，若被人恶意写入违法信息还可能会侵害您的合法权益。除特殊场景外，不建议您配置此权限。 • <i>public-read-write</i>：任何人（包括匿名访问者）都可以对该Bucket内文件进行读写操作。这有可能造成您数据的外泄以及费用激增，请谨慎操作。
--maxupspeed	最大上传速度，单位为KB/s。默认值为0，表示不限制上传速度。
--disable-crc64	关闭CRC64数据校验。
--payer	请求的支付方式。如果希望访问指定路径下的资源产生的流量、请求次数等费用由请求者支付，请将此选项的值设置为 <i>requester</i> 。
-j, --job	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。
--parallel	单文件操作时的并发任务数，取值范围为1-10000。如果不设置此选项，默认由ossutil根据操作类型和文件大小自行决定。
--retry-times	发生错误后的重试次数。默认值为10，取值范围为1~500。
--tagging	文件的标签信息，格式为 <code>TagkeyA=TagvalueA&TagkeyB=TagvalueB....</code> 。

使用示例

某阿里云账号下有examplebucket1和examplebucket2两个Bucket，examplebucket1下有 *exampledir*和*srcdir*两个文件夹。examplebucket2下有 *destdir*文件夹。文件结构如下：

```
examplebucket1      examplebucket2
├─ exampledir/      └─ destdir/
│   └─ a.txt         └─ c.txt
│   └─ b.txt         └─ e.txt
└─ srcdir/
   └─ d.txt
```

示例场景及命令如下：

- 将examplebucket1的*exampledir*文件夹同步到*srcdir*文件夹内。

```
./ossutil64 sync oss://examplebucket1/exampledir/ oss://examplebucket1/srcdir/
```

命令执行完毕后，*exampledir*文件夹内的所有文件被同步到*srcdir*文件夹下。

```
examplebucket1          examplebucket1
├─ exampledir/          └─ exampledir/
│   └─ a.txt            │   └─ a.txt
│       └─ b.txt        │       └─ b.txt
└─ srcdir/              └─ srcdir/
    └─ d.txt            │   └─ a.txt
                        │       └─ b.txt
                        │       └─ d.txt
```

- 将examplebucket1的*exampledir*文件夹同步到examplebucket2的*destdir*文件夹。

```
./ossutil64 sync oss://examplebucket1/exampledir/ oss://examplebucket2/destdir/
```

命令执行后，examplebucket1的*exampledir*文件夹中所有文件被同步到examplebucket2的*destdir*文件夹。

```
examplebucket1          examplebucket2
├─ exampledir/          └─ destdir/
│   └─ a.txt            │   └─ a.txt
│       └─ b.txt        │   └─ b.txt
└─ srcdir/              └─ c.txt
    └─ d.txt            └─ e.txt
```

- 结合--delete选项将examplebucket1的所有文件同步到examplebucket2，并删除examplebucket2存在而examplebucket1不存在的文件。

```
./ossutil64 sync oss://examplebucket1 oss://examplebucket2 --delete
```

命令执行后，examplebucket1的*exampledir*文件夹将同步到examplebucket2，examplebucket2内已有的文件*c.txt*和*e.txt*将被删除。

```
examplebucket1          examplebucket2
├─ exampledir/          └─ exampledir/
│   └─ a.txt            │   └─ a.txt
│       └─ b.txt        │   └─ b.txt
└─ srcdir/              └─ srcdir/
    └─ d.txt            │   └─ d.txt
```

- 以上示例同步成功后，返回结果中将包含同步的文件数量、文件大小以及完成同步操作所用时长，示例如下：

```
Succeed: Total num: 2, size: 750,081. OK num: 2(upload 2 files).
average speed 1641000(byte/s)
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如，您需要将另一个阿里云账号下，华东2（上海）地域下源存储空间examplebucket的文件夹srcfolder同步至目标存储空间testbucket的文件夹examplefolder，命令如下：

```
./ossutil64 sync oss://examplebucket/srcfolder/ oss://testbucket/examplefolder/ -e oss-cn-shanghai.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA**** -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.40. update（ossutil版本升级）

update命令用于将ossutil升级至最新版本。

 **注意** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

```
./ossutil64 update [-f]
```

该命令用于检查并输出ossutil当前版本与最新版本。如果检测到ossutil有更新版本，则询问是否进行升级。如果指定了-f选项，当有可用更新版本时不再询问，直接升级。

使用示例

- 询问是否升级至ossutil最新版本

```
./ossutil64 update
当前版本为：v1.5.1，最新版本为：1.7.5
确定更新版本(y or N)? y
```

- 强制升级至ossutil最新版本

```
./ossutil64 update -f
```

2.3.41. website（静态网站托管及回源配置）

静态网站托管功能允许您将静态网站托管到OSS的存储空间（Bucket），并使用Bucket的访问域名访问这个网站。Bucket配置静态网站托管后，当您的用户向OSS请求的数据不存在时，可通过设置回源规则确保其仍然可以获取正确的数据。**website**命令用于添加、修改、查询、删除Bucket的静态网站托管以及回源配置。

注意事项

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- Bucket配置静态网站托管后，当您的用户向OSS请求的数据不存在时，本应返回404错误。如果您设置了回源规则，并填写了数据的正确地址，则用户可以通过回源规则从OSS获取到正确的数据。

有关静态网站托管功能的更多信息，请参见[静态网站托管介绍](#)。

有关回源功能的更多信息，请参见[回源类型概述](#)。

- 当您在OSS ON云盒中使用该命令时：

- i. 将配置文件中的Endpoint替换为云盒Endpoint。更多信息，请参见[云盒Endpoint](#)。
- ii. 在本文已有示例的基础上添加 `--sign-version`、`--region`以及`--cloudbox-id`选项。关于这三个选项的具体用法，请参见[通用选项](#)。

添加或修改Website配置

如果Bucket此前未添加过Website配置，则使用该命令将为Bucket添加Website配置。如果Bucket此前已添加Website配置，则使用该命令将直接覆盖已有的Website配置。

添加或修改Website配置步骤如下：

- 1. 创建本地文件，并在文件中添加Website配置。
- 2. 将Website配置添加到目标Bucket。

添加或修改Website配置的命令格式及示例如下：

● 命令格式

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

参数说明如下：

参数	说明
bucketname	待添加或修改Website配置的Bucket名称。
local_xml_file	配置Website规则的本地文件名称，例如 <code>localfile.txt</code> 。

● 使用示例

- i. 在本地创建名为 `localfile.txt` 文件，并根据使用场景配置Website规则。

如下为Website的完整配置参考示例，示例中包含了静态网站托管以及回源配置。您也可以根据具体的业务场景需求，选择仅添加静态网站托管或者回源配置。

■ 静态网站托管配置

配置示例中通过<IndexDocument>字段指定静态页面的默认首页为 `index.html`，<ErrorDocument>字段指定静态页面的默认404页为 `error.html`。

■ 回源配置

配置示例中通过<RoutingRules>字段指定回源规则。有关回源规则中包含的各参数配置详情，请参见[设置回源规则](#)。

```
<?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>
      </Condition>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```

```

        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
    </Condition>
    <Redirect>
        <RedirectType>Mirror</RedirectType>
        <PassQueryString>true</PassQueryString>
        <MirrorURL>http://example.com/</MirrorURL>
        <MirrorPassQueryString>true</MirrorPassQueryString>
        <MirrorFollowRedirect>true</MirrorFollowRedirect>
        <MirrorCheckMd5>false</MirrorCheckMd5>
        <MirrorHeaders>
            <PassAll>true</PassAll>
            <Pass>myheader-key1</Pass>
            <Pass>myheader-key2</Pass>
            <Remove>myheader-key3</Remove>
            <Remove>myheader-key4</Remove>
            <Set>
                <Key>myheader-key5</Key>
                <Value>myheader-value5</Value>
            </Set>
        </MirrorHeaders>
    </Redirect>
</RoutingRule>
<RoutingRule>
    <RuleNumber>2</RuleNumber>
    <Condition>
        <KeyPrefixEquals>abc</KeyPrefixEquals>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
        <IncludeHeader>
            <Key>host</Key>
            <Equals>test.oss-cn-beijing-internal.aliyuncs.com</Equals>
        </IncludeHeader>
    </Condition>
    <Redirect>
        <RedirectType>AliCDN</RedirectType>
        <Protocol>http</Protocol>
        <HostName>example.com</HostName>
        <PassQueryString>false</PassQueryString>
        <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith>
        <HttpRedirectCode>301</HttpRedirectCode>
    </Redirect>
</RoutingRule>
</RoutingRules>
</WebsiteConfiguration>

```

ii. 为examplebucket添加Website配置。

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

以下输出结果表明已成功添加Website配置。

```
0.299514(s) elapsed
```

获取website配置

- 命令格式


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

参数说明如下：

参数	说明
bucketname	待获取Website配置的目标Bucket名称。
local_xml_file	用于存放Website配置的本地文件名称，例如 <code>localfile.txt</code> 。如果未指定此参数，则Website配置将直接输出到屏幕。

● 使用示例

- 获取目标存储空间examplebucket的Website配置，并将配置结果写入本地文件 `localfile.txt`。

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

以下输出结果表明已成功获取Website配置，并将其写入本地localfile.txt文件。

```
0.212407(s) elapsed
```

- 获取目标存储空间examplebucket的Website配置，并将配置结果直接输出屏幕。

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

以下输出结果表明已成功获取Website配置。

```
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
    <SupportSubDir>>false</SupportSubDir>
    <Type>0</Type>
  </IndexDocument>
  <ErrorDocument>
    <Key>error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <RuleNumber>1</RuleNumber>
      <Condition>
        <KeyPrefixEquals>abc/</KeyPrefixEquals>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <RedirectType>Mirror</RedirectType>
        <PassQueryString>true</PassQueryString>
        <MirrorURL>http://example.com/</MirrorURL>
        <MirrorPassQueryString>true</MirrorPassQueryString>
        <MirrorPassOriginalSlashes>>false</MirrorPassOriginalSlashes>
        <MirrorFollowRedirect>true</MirrorFollowRedirect>
        <MirrorIsExpressTunnel>>false</MirrorIsExpressTunnel>
        <MirrorUserLastModified>>false</MirrorUserLastModified>
        <MirrorCheckMd5>>false</MirrorCheckMd5>
        <MirrorSwitchAllErrors>>false</MirrorSwitchAllErrors>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```

```

    <MirrorHeaders>
      <PassAll>true</PassAll>
      <Pass>myheader-key1</Pass>
      <Pass>myheader-key2</Pass>
      <Remove>myheader-key3</Remove>
      <Remove>myheader-key4</Remove>
      <Set>
        <Key>myheader-key5</Key>
        <Value>myheader-value5</Value>
      </Set>
    </MirrorHeaders>
    <MirrorUsingRole>false</MirrorUsingRole>
    <MirrorAllowHeadObject>false</MirrorAllowHeadObject>
    <EnableReplacePrefix>false</EnableReplacePrefix>
  </Redirect>
</RoutingRule>
<RoutingRule>
  <RuleNumber>2</RuleNumber>
  <Condition>
    <IncludeHeader>
      <Key>host</Key>
      <Equals>test.oss-cn-beijing-internal.aliyuncs.com</Equals>
    </IncludeHeader>
    <KeyPrefixEquals>abc/</KeyPrefixEquals>
    <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
  </Condition>
  <Redirect>
    <RedirectType>AliCDN</RedirectType>
    <Protocol>http</Protocol>
    <HostName>example.com</HostName>
    <PassQueryString>false</PassQueryString>
    <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith>
    <EnableReplacePrefix>false</EnableReplacePrefix>
    <HttpRedirectCode>301</HttpRedirectCode>
  </Redirect>
</RoutingRule>
</RoutingRules>
</WebsiteConfiguration>
0.157648(s) elapsed

```

删除Website配置

- 命令格式

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

- 使用示例

删除目标存储空间examplebucket的Website配置。

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

以下输出结果表明已成功删除Website配置。

```
0.212409(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）名为testbucket的Bucket添加Website配置，命令如下：

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

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.3.42. worm（合规保留策略）

对象存储OSS支持基于时间的合规保留策略，您可以在合规保留策略的保留周期内以“不可删除、不可篡改”的方式保存和使用OSS数据。本文介绍如何使用worm命令管理Bucket的合规保留策略。

说明

- 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。
- 关于合规保留策略的更多信息，请参见[合规保留策略](#)。

创建并锁定合规保留策略

若您希望使用合规保留策略保护您Bucket中的文件，您需要为Bucket创建一条合规保留策略，然后锁定该策略。

1. 创建合规保留策略

命令格式

```
./ossutil64 worm init oss://BucketName days
```

参数说明：

参数	说明
BucketName	配置合规保留策略的目标Bucket名称。
days	指定合规保留策略的保留周期。在保留周期内，您无法对目标Bucket中的Object进行任何修改和删除操作。 - 单位：天 - 取值范围：1~25550

使用示例

为examplebucket创建合规保留策略，并指定其保留周期为180天。

```
./ossutil64 worm init oss://examplebucket 180
```

以下输出结果表明已成功创建合规保留策略。

```
init success, worm id is 581D8A7FFA064C80827CAB4076A93A78
```

2. 锁定合规保留策略

◦ 命令格式

```
./ossutil64 worm complete oss://BucketName WormId
```

参数说明：

参数	说明
BucketName	指定要锁定合规保留策略的目标Bucket名称。
WormId	填写创建合规保留策略时返回的策略ID。

◦ 使用示例

锁定examplebucket的合规保留策略。

```
./ossutil64 worm complete oss://examplebucket 581D8A7FFA064C80827CAB4076A93A78
```

以下输出结果表明已锁定合规保留策略。

```
0.073810(s) elapsed
```

延长保留周期

合规保留策略锁定后，在保留周期未到期之前，任何人都无法修改和删除Bucket中的Object。若当前的保留周期无法满足数据保护的需求，您可以通过以下命令延长保留周期。

● 命令格式

```
./ossutil64 worm extend oss://BucketName days WormId
```

● 使用示例

将examplebucket的合规保留策略的保留周期延长到360天。

```
./ossutil64 worm extend oss://examplebucket 360 581D8A7FFA064C80827CAB4076A93A78
```

以下输出结果表明保留周期已延长到360天。

```
0.067810(s) elapsed
```

查询合规保留策略配置

对于已创建的合规保留策略，您可以通过以下命令查询配置参数。

● 命令格式

```
./ossutil64 worm get oss://BucketName
```

● 使用示例

查询examplebucket的合规保留策略。

```
./ossutil64 worm get oss://examplebucket
```

以下输出结果表明已查询到合规保留策略的配置参数，结果中包含策略ID、状态、保留天数、策略创建时间。

```
<WormConfiguration>
  <WormId>581D8A7FFA064C80827CAB4076A93A78</WormId>
  <State>Locked</State>
  <RetentionPeriodInDays>360</RetentionPeriodInDays>
  <CreationDate>2021-01-19T03:36:53.000Z</CreationDate>
</WormConfiguration>
```

删除合规保留策略

在合规保留策略未锁定前，您可以删除该策略。

- 命令格式

```
./ossutil64 worm abort oss://BucketName
```

- 使用示例

删除为examplebucket配置的合规保留策略。

```
./ossutil64 worm abort oss://examplebucket
```

以下输出结果表明策略已删除。

```
0.067810(s) elapsed
```

通用选项

当您需要通过命令行工具ossutil切换至另一个地域的Bucket时，可以通过-e选项指定该Bucket所属的Endpoint。当您需要通过命令行工具ossutil切换至另一个阿里云账号下的Bucket时，可以通过-i选项指定该账号的AccessKey ID，并通过-k选项指定该账号的AccessKey Secret。

例如您需要为另一个阿里云账号下，华东1（杭州）名为test的Bucket创建合规保留策略，命令如下：

```
./ossutil64 worm init oss://test -e oss-cn-hangzhou.aliyuncs.com -i LTAI4Fw2NbDUCV8zYUzA***
* -k 67DLVBkH7EamOjy2W5RVAHUY9H****
```

关于此命令的其他通用选项的更多信息，请参见[通用选项](#)。

2.4. 查看选项

您可以通过-h命令查看ossutil支持的所有选项。

 **说明** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的./ossutil64替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

命令格式

```
./ossutil64 -h
```

若要查看某个命令支持哪些选项，请使用./ossutil64 help [command]命令来查看，例如./ossutil64 help cp。

通用选项

以下为ossutil的通用选项，这些选项可以在大部分命令中使用：

选项	说明
-c, --config-file	ossutil工具的配置文件路径，ossutil启动时将从配置文件读取配置。当您需要管理多个账号下的Bucket时，可以生成多个配置文件，并将其中一个指定为默认配置文件。当您需要管理其他账户下的Bucket时，请通过-c指定正确的配置文件。
-e, --endpoint	指定Bucket对应的Endpoint，当您需要管理多个地域的Bucket时，可以通过此选项指定多个Endpoint。各地域Endpoint详情请参见 访问域名和数据中心 。
-i, --access-key-id	指定访问OSS使用的AccessKey ID，当您需要管理多个账号下的Bucket时，可通过此选项指定对应的AccessKey ID。
-k, --access-key-secret	指定访问OSS使用的AccessKey Secret，当您需要管理多个账号下的Bucket时，可通过此选项指定对应的AccessKey Secret。
-p, --password	指定访问OSS使用的AccessKey Secret，输入该选项时会提示用户从键盘输入AccessKey Secret，ossutil工具以从键盘读取的AccessKey Secret为准，并忽略通过其他方式配置的AccessKey Secret。
--loglevel	在当前工作目录下输出ossutil日志文件ossutil.log。该选项默认为空，表示不输出日志文件。 取值： • info：用于打印ossutil操作记录。 <pre>./ossutil64 [command] --loglevel=info</pre> • debug：可以输出HTTP流水日志以及原始签名串信息，方便定位问题。 <pre>./ossutil64 [command] --loglevel=debug</pre>
--proxy-host、--proxy-user、--proxy-pwd	在代理上网环境下，需指定如下三个选项： • --proxy-host：网络代理服务器的URL地址，支持HTTP、HTTPS以及SOCKS5。 • --proxy-user：网络代理服务器的用户名，默认为空。 • --proxy-pwd：网络代理服务器的密码，默认为空。 通过这三个选项指定代理服务器信息后，ossutil将使用您指定的信息并通过代理服务器访问OSS。 <pre>./ossutil64 ls oss://bucket1 --proxy-host http://47.88.**.*:3128 --proxy-user test --proxy-pwd test</pre>

选项	说明
--mode	凭证类型。取值如下： - AK: 使用AccessKey ID和AccessKey Secret的方式访问。 - StsToken: 使用STS Token的方式访问。 - RamRoleArn: 使用RAM用户的AssumeRole的方式访问。 - EcsRamRole: 在ECS实例上通过EcsRamRole实现免密验证。 如果不添加此选项，默认按照原有的鉴权逻辑处理。
--ecs-role-name	EcsRamRole鉴权模式下的角色名称。
--token-timeout	RamRoleArn鉴权模式下AssumeRole参数中指定的临时访问凭证Token的有效时间。单位为秒，默认值为3600。
--ram-role-arn	RamRoleArn鉴权模式下的RAM角色ARN。
--role-session-name	RamRoleArn鉴权模式下的会话名称。
--read-timeout	客户端读超时的时间，单位为秒，默认值为1200。
--connect-timeout	客户端连接超时的时间，单位为秒，默认值为120。
--sts-region	STS服务的接入地域，格式为 <code>cn-hangzhou</code>。STS服务支持的所有接入地域的详情，请参见接入地址。 如果不添加此选项，RamRoleArn鉴权模式下指定的sts endpoint 为 <code>sts.aliyuncs.com</code>。
--skip-verify-cert	不校验服务端的数字证书。
--ua	此选项用于指定HTTP Header中的User Agent值。指定后，该值将自动添加到ossutil工具设置的User Agent值的末尾。如果您希望在日志中查看完整的User Agent值，请结合使用 <code>--loglevel debug</code> 选项。
--sign-version	签名版本，取值为v4。仅OSS ON云盒场景中需要使用该选项。
--region	云盒Bucket所在地域。仅OSS ON云盒场景中需要使用该选项。
--cloudbox-id	云盒ID。仅OSS ON云盒场景中需要使用该选项。

其他选项

除以上通用选项外，ossutil各命令包含的其他选项如下：

选项名称	描述
-s, --short-format	显示精简格式，如果未指定该选项，默认显示长格式。

选项名称	描述
<code>--bigfile-threshold</code>	开启大文件断点续传的文件大小阈值，单位为Byte，默认值为100 MByte，取值范围为0~9223372036854775807。
<code>--acl</code>	配置读写权限ACL。
<code>--range</code>	下载文件时，可以指定文件内容的字节范围进行下载，字节从0开始编号。 - 可以指定一个区间，例如 <code>3-9</code> 表示从第3个字节到第9个字节（包含第3和第9字节）。 - 可以指定从什么字段开始，例如 <code>3-</code> 表示从第3个字节开始到文件结尾（包含第3个字节）。 - 可以指定从什么字段结束，例如 <code>-9</code> 表示从0字节到第9个字节（包含第9个字节）。
<code>--all-versions</code>	表示Object所有版本。
<code>--type</code>	数据校验的方式。取值如下： <code>crc64</code>（默认值）：数据CRC64校验。 <code>md5</code>：数据MD5校验。
<code>-v, --version</code>	显示ossutil的版本并退出。
<code>-u, --update</code>	更新操作。
<code>--origin</code>	HTTP请求头origin字段的值。
<code>--upmode</code>	<code>probe</code>命令中使用的上传模式。取值如下： <code>normal</code>（默认值）：简单上传。 <code>append</code>：追加上传。 <code>multipart</code>：分片上传。
<code>--sse-algorithm</code>	Bucket的加密方式。取值如下： <code>KMS</code>：使用KMS托管密钥进行加解密，即SSE-KMS。 <code>AES256</code>：使用OSS完全托管密钥进行加解密，即SSE-OSS。
<code>--include</code>	列举符合指定条件的Object。例如 <code>*.jpg</code> 表示列举所有格式为JPG的文件。
<code>--exclude</code>	列举不符合指定条件的Object。例如 <code>*.txt</code> 表示列举所有格式不为TXT的文件。
<code>-r, --recursive</code>	递归操作。当指定该选项时，ossutil会对Bucket下所有符合条件的Object进行操作，否则只对URL中指定的单个Object进行操作。
<code>--addr</code>	增加此项并填写正确的网络地址，ossutil会通过ping操作验证本地到目标地址的网络连通性。 默认值：<code>www.aliyun.com</code>
<code>--kms-masterkey-id</code>	KMS密钥托管服务中的主密钥ID。

选项名称	描述
<code>--version-id</code>	Object的版本ID。
<code>--version-id-marker</code>	列举Version ID字母排序在marker之后的Object版本。仅当Bucket开启版本控制后 可用。
<code>-m, --multipart</code>	指定操作的对象为Bucket中未完成的Multipart事件。
<code>-d, --directory</code>	仅返回当前目录下的文件和子目录。
<code>--payer</code>	请求的支付方式，如果为请求者付费模式，可以将该值设置为 <code>requester</code> 。
<code>--maxuploadspeed</code>	最大上传速度，单位为KB/s，默认值为0（不受限制）。
<code>--maxdownloadspeed</code>	最大下载速度，单位为KB/s，默认值为0（不受限制）。
<code>--retry-times</code>	发生错误后的重试次数。默认值为10，取值范围为1~500。
<code>--download</code>	通过使用文件URL下载目标文件到本地的方式探测本地和目标Bucket之间的网络状态。
<code>-j, --jobs</code>	多文件操作时的并发任务数，默认值为3，取值范围为1~10000。
<code>-a, --all-type</code>	指定操作的对象为Bucket中的Object和未完成的Multipart事件。
<code>--disable-empty-referer</code>	表示不允许referer字段为空。
<code>--method</code>	表示HTTP的请求类型，取值为PUT、GET、DELETE等。
<code>--output-dir</code>	指定输出文件所在的目录，输出文件目前包含cp命令批量拷贝文件出错时所产生的 report文件。 默认值：当前目录下的 <code>ossutil_output</code> 目录。
<code>--meta</code>	设置Object的meta，格式为[header: value#header: value...]，如： <code>Cache-Control: no-cache#Content-Encoding: gzip</code> 。
<code>--object</code>	用于probe命令中指定文件名称。
<code>--end-time</code>	Linux或Unix系统下的时间戳。如果使用该选项，则最后更新时间晚于通过此选项指定时间的Object会被忽略。
<code>--limited-num</code>	返回结果的最大个数。
<code>-L, --language</code>	设置ossutil工具的语言。取值如下： - CH（默认值）：中文。设置为CH时，请确保您的系统编码方式为UTF-8。 - EN：英文。
<code>--delete</code>	删除Bucket、Object或碎片操作。
<code>-b, --bucket</code>	仅在删除Bucket时使用此选项。

选项名称	描述
<code>--disable-crc64</code>	关闭CRC64校验。默认情况下，ossutil进行数据传输会打开CRC64校验。
<code>--upload</code>	通过将本地文件上传到目标Bucket的方式来探测本地和目标Bucket之间的网络状态。
<code>--part-size</code>	分片大小，单位为字节，默认情况下ossutil根据文件大小自行计算合适的分片大小值。如果需要性能调优时或者其他特殊需求时，可以设置该值，取值范围为1~9223372036854775807。
<code>--timeout</code>	签名URL的超时时间，单位为秒，默认值为60，取值范围为0~9223372036854775807。
<code>--checkpoint-dir</code>	指定断点续传记录信息所在的目录。断点续传操作失败时，ossutil会自动创建名为 <code>.ossutil_checkpoint</code> 的目录，并在该目录下记录checkpoint信息，断点续传成功后会删除该目录。如果通过该选项指定了目录，请确保指定的目录可以被删除。
<code>--url</code>	文件URL。
<code>--marker</code>	列举名称字母排序在marker之后的Bucket、Object或者Part。
<code>-f, --force</code>	强制操作，不进行询问提示。
<code>--snapshot-path</code>	指定保存上传、下载文件时的快照信息所在的目录。在下次上传或下载文件时，ossutil会读取指定路径下的快照信息进行增量上传或下载。 • <code>--snapshot-path</code>选项用于在某些场景下加速增量上传、下载多个适用于文件（拷贝操作不支持该选项）。例如文件数量较多且两次上传期间没有其他用户更改OSS上对应的Object。 • <code>--snapshot-path</code>选项通过在本地记录成功上传、下载的文件的本地的lastModifiedTime，从而在下次上传、下载时通过比较lastModifiedTime来决定是否跳过相同文件，所以在使用该选项时，请确保两次上传、下载期间没有其他用户更改了OSS上的对应Object。当不满足该场景时，如果想要增量上传、下载批量文件，请使用<code>--update</code>选项。 • ossutil不会主动删除snapshot-path下的快照信息，为了避免快照信息过多，当您确定快照信息无用时，请自行清理snapshot-path。 • 由于读写snapshot信息需要额外开销，当要批量上传、下载的文件数比较少或网络状况比较好或有其他用户操作相同Object时，并不建议使用该选项。可以使用<code>--update</code>选项来增量上传/下载。 • <code>--update</code>选项和<code>--snapshot-path</code>选项可以同时使用，ossutil会优先根据snapshot-path信息判断是否跳过此文件，如果不满足跳过条件，再根据<code>--update</code>判断是否跳过此文件。
<code>--start-time</code>	值为Linux或Unix系统下面的时间戳，如果输入这个选项，最后更新时间早于该时间的Object会被忽略。

选项名称	描述
<code>--storage-class</code>	设置Object的存储方式。取值如下： <code>Standard</code>（默认值）：适用于频繁的数据访问。 <code>IA</code>：适用于较低访问频率（平均每月访问频率1到2次）的业务场景，有最低存储时间（30天）和最小计量单位（64 KB）要求。支持数据实时访问，访问数据时会产生数据取回费用。 <code>Archive</code>：适用于数据长期保存的业务场景，有最低存储时间（60天）和最小计量单位（64 KB）要求。数据需解冻（约1分钟）后访问，解冻会产生数据取回费用。 <code>ColdArchive</code>：适用于需要超长时间存放的极冷数据，有最低存储时间（180天）和最小计量单位（64 KB）要求。数据需解冻后访问，解冻时间根据数据大小和选择的解冻模式决定，解冻会产生数据取回费用。
<code>-t, --sts-token</code>	访问OSS使用的STSToken，仅在使用STS临时授权方式访问OSS时需要配置此项。该选项值会覆盖配置文件中的相应设置。stsToken生成方式请参见 临时访问凭证 。
<code>--parallel</code>	单文件内部操作的并发任务数，取值范围为1~10000，默认由ossutil根据操作类型和文件大小自行决定。
<code>--partition-download</code>	用于分区下载。该选项取值格式为 <code>分区编号:总分区数</code> 。例如1: 5，表示当前ossutil下载分区1，总共有5个分区。分区号从1开始编号，Object的分区规则由工具内部算法决定。通过该选项将待下载的Object分成多个分区，并由多个ossutil命令并行下载，各个ossutil命令下载各自的分区。
<code>--bucketname</code>	Bucket名称。
<code>--encoding-type</code>	对 <code>oss://bucket_name</code> 后面的key进行编码，取值为 <code>url</code> 。如果不指定该选项，则表示key未经过编码。
<code>--origin</code>	HTTP请求头Origin字段的值，表示请求来源域，用来标识跨域请求。
<code>--acr-method</code>	HTTP请求头Access-Control-Request-Method的值，取值为GET、PUT、POST、DELETE、HEAD。
<code>--acr-headers</code>	HTTP请求头Access-Control-Request-Headers的值，表示在实际请求中除简单头部以外需要用到的Header。如果有多个Header，各个Header之间用英文的逗号（,）分隔，再加上双引号。例如 <code>--acr-headers "header1,header2,header3"</code> 。
<code>--upload-id-marker</code>	列举Upload ID字母排序在Marker之后的Part。
<code>-h, --help</code>	显示帮助信息。
<code>--traffic-limit</code>	<code>sign</code>命令中用于指定HTTP的访问速度。单位为bit/s。 默认值：0，表示不限速。 取值范围：819200~838860800，即100 KB/s~100 MB/s。
<code>--local-host</code>	<code>cp</code> 命令中用于指定ossutil工具的本地IP地址。指定该选项后，ossutil将通过指定IP地址访问OSS。

选项名称	描述
<code>--enable-symlink-dir</code>	上传链接子目录。默认不上传。 <code>probe</code> 命令可以探测是否存在死循环链接文件或者目录。
<code>--only-current-dir</code>	仅上传、下载、拷贝当前目录下的文件，忽略子目录及子目录下的文件。
<code>--disable-dir-object</code>	不为上传的目录生成Object，但您仍可以在OSS控制台看到对应的目录结构。当您删除目录内的文件时，该目录也会消失。
<code>--probe-item</code>	表示probe命令的探测项目，取值如下： • <code>upload-speed</code>: 探测上传带宽。 • <code>download-speed</code>: 探测下载带宽。 • <code>cycle-symlink</code>: 探测本地文件目录中是否存在死循环链接文件或者目录。
<code>--redundancy-type</code>	Bucket的数据容灾类型。取值如下： • <code>LRS</code>（默认值）：本地冗余LRS将您的数据冗余存储在同一个可用区的不同存储设备上，可支持两个存储设备并发损坏时，仍维持数据不丢失，可正常访问。 • <code>ZRS</code>：同城冗余ZRS采用多可用区（AZ）机制，将您的数据冗余存储在同一地域（Region）的3个可用区。可支持单个可用区（机房）整体故障时（如断电、火灾等），仍然能够保障数据的正常访问。
<code>--disable-encode-slash</code>	表示不对URL中的正斜线（/）进行编码。
<code>--disable-all-symlink</code>	上传时忽略所有的符号链接子文件以及符号链接子目录。
<code>--tagging</code>	上传或复制文件时设置文件的对象标签，格式为 <code>"abc=1&bcd=2&....."</code> 。
<code>--disable-ignore-error</code>	批量操作时不忽略错误。
<code>--block-size</code>	定义输出结果中指定Bucket或目录下包含的Object大小，取值为KB、MB、GB或TB。不添加此选项时，默认以Byte为单位统计Object的大小。  注意 此选项仅在ossutil1.7.3及以上版本可用。

2.5. 常见问题

本文介绍在使用ossutil时可能出现的问题及处理方法。

 **说明** 本文各命令行示例均基于Linux 64位系统，其他系统请将命令开头的`./ossutil64`替换成对应的Binary名称。详情请参见[命令行工具ossutil快速入门](#)。

使用-u参数上传文件时出现skip提示

问题分析：使用-u参数上传文件的时候，ossutil会将上传文件和存储空间（Bucket）内已有文件进行一次比对。若发现上传的文件与目标Bucket内已有文件同名，且该文件的最后修改时间早于或等于Bucket内已有文件，上传时会跳过该文件；若该文件的最后修改时间晚于Bucket内已有文件，则重新上传该文件。所以使用-u参数上传文件时出现skip提示是正常现象。

解决方案：确认Bucket内文件上传无问题后忽略该提示。

文件解冻时出现403报错

问题分析：操作解冻文件的过程中出现403，有以下两种可能性。

- 若您使用子账号操作文件，可能是子账号权限不足，无目标文件的操作权限。
- 您文件内容违禁OSS被封禁了。

解决方案：

- 子账号权限不足：增加子账号权限。
- 文件内容违禁：删除或忽略该文件。

使用ls命令查看Bucket无文件，删除Bucket时报错

问题分析：使用ls命令列举Bucket内的文件且未携带任何选项时，无法列举碎片、历史版本文件（仅存在于开启过版本控制的Bucket）。若Bucket为非空（即Bucket中存在碎片、历史版本文件），仅使用rm命令无法删除该Bucket。

解决方案：

- 先删除碎片和历史版本文件（未开启过版本控制请忽略），再删除Bucket。
 - i. 删除碎片和历史版本文件。

- 列举并删除碎片：

```
./ossutil64 ls oss://bucket1 -m
./ossutil64 rm -m oss://bucket1 -r
```

- 列举并删除历史版本文件：

```
./ossutil64 ls oss://bucket1 --all-versions
./ossutil64 rm oss://bucket1 --all-versions -r
```

- ii. 删除Bucket。

```
./ossutil64 rm oss://bucket1 -b
```

- 强制删除Bucket。
 - 若存储空间未开启版本控制，使用如下命令强制删除Bucket：

```
./ossutil64 rm oss://bucketname -abrf
```

- 若存储空间已开启版本控制，使用如下命令强制删除Bucket：

```
./ossutil64 rm oss://bucketname -abrf --all-versions
```



警告 Bucket及文件被删除后，无法恢复，请谨慎使用该命令。

文件上传、下载进度超过100%

问题分析：ossutil在上传、下载文件时，会自动生成一个名为`.ossutil_checkpoint`文件夹。当目标文件超过100 MB时，ossutil默认使用断点续传上传或下载目标文件，并将过程中生成的断点信息文件保存在`.ossutil_checkpoint`文件夹中。上传、下载任务完成后，会自动删除这个文件夹。若单机运行超过一个ossutil实例，且都在进行上传或下载任务，当其中一个ossutil的任务完成后，会自动删除`.ossutil_checkpoint`文件夹，导致其它需要使用断点续传的ossutil任务进度超过100%且无法完成。

解决方案：

- 将当前任务结束，重新开始上传、下载任务。
- 在cp命令中加上`--checkpoint-dir`参数，并手动指定一个与默认checkpoint文件夹不同名的文件夹。例如：

```
./ossutil64 cp oss://bucket1/myphoto.jpg /dir --checkpoint-dir checkpoint
```

使用ossutil发起单个文件的下载请求，但OSS日志里面有多条请求记录

问题分析：出现该问题有两种可能。

- 请求异常或请求失败时，ossutil会进行重试操作，默认为10次。
- 如果您下载的文件超过100 MB，ossutil会使用range方式去请求，即每次请求该文件的部分内容，通过多次请求完成完整文件的下载。因此，超过100 MB的文件下载时也会产生多条请求记录。

使用ossutil执行相关命令时，报错 `Error: accessKeyId and ecsUrl are both empty`

问题分析：未在home目录下生成配置文件。

解决方案：

- 使用ossutil执行相关命令之前，需在当前用户的home目录下生成配置文件。具体步骤，请参见[下载和安装](#)。
- 使用ossutil执行相关命令时单独指定配置文件路径。例如，通过cp命令上传文件时指定配置文件路径：

```
./ossutil64 cp examplefile.txt oss://examplebucket/destfolder/ -c /home/admin/.ossutilconfig
```

3.图形化管理工具ossbrowser

3.1. 安装并登录ossbrowser

ossbrowser是阿里云官方提供的OSS图形化管理工具，提供类似Windows资源管理器的功能。本文介绍如何快速安装并登录ossbrowser。

操作步骤

1. 下载并安装ossbrowser。

当前版本	操作系统	下载地址	说明
1.16.0	Windows x32	oss-browser-win-x32.zip	支持Windows 7及以上版本。 如果遇到安装ossbrowser后无法正常使用的 问题，请参见 Windows系统安装某软件后无法 打开ossbrowser 。
	Windows x64	oss-browser-win-x64.zip	❓ 说明 不推荐通过Windows Server 安装ossbrowser。如果您在使用 Windows Server安装ossbrowser遇到问 题时，请自行构建源码解决兼容性问题。
	macOS	oss-browser-macos-x64.zip	macOS系统默认无法打开未验证的开发者应 用。如果出现该问题，请参见 macOS系统无法 打开ossbrowser 修改系统安全设置。
	Linux x64（不推荐）	oss-browser-linux-x64.zip	如果在Linux系统安装ossbrowser遇到问题 时，请自行下载 源码 编译排查。

2. 以Windows系统为例，按如下步骤登录ossbrowser。

- i. 双击打开 `oss-browser.exe` 。
- ii. 选择以下任意一种方式登录ossbrowser。
 - 通过AK登录

您可以通过阿里云账号或RAM用户的AccessKey（AK）信息登录ossbrowser。

AK登录

授权码登录

* Endpoint: ?

默认 (公共云)

▼

☒ HTTPS加密

* AccessKeyId:

LTAI4GBnnmywkG

* AccessKeySecret:

.....

预设OSS路径: ?

oss://examplebucket/exampleobject.txt

☐ 请求者付费模式 ?

区域:

华东1(杭州)

▼

备注:

可以为空, 最多30个字

☒ 保持登录

☒ 记住秘钥 ?

登录

AK历史

通过AK登录时，需按如下说明完成各配置项。

参数	说明
Endpoint	选择登录的访问域名。 ■ 默认（公共云）：使用目标Bucket所在地域对应的Endpoint登录。选择此种登录方式时，可选中HTTPS加密对传输过程加密。 注意 通过该方式登录时，不支持访问华北6（乌兰察布）和菲律宾（马尼拉）地域的Bucket。如果您希望访问这两个地域的Bucket，需选择自定义的登录方式。 ■ 自定义：使用自定义Endpoint的方式登录，例如 <code>oss-cn-wulanchabu.aliyuncs.com</code>。选择此种登录方式时，可使用公共云下OSS的任意Endpoint。有关Region与Endpoint的对应关系，请参见访问域名和数据中心。 ■ cname：如果您希望通过自有域名访问OSS资源，请先绑定自有域名。具体操作，请参见绑定自定义域名。然后选择此项，填写绑定的自有域名。
AccessKeyId、AccessKeySecret	填写账号的AccessKey（AK）信息。获取AK的方式，请参见获取AccessKey。 注意 为保证数据安全，推荐您使用RAM用户的AK登录ossbrowser。使用RAM用户登录之前，需要为RAM用户配置 <code>AliyunOSSFullAccess</code>、<code>AliyunRAMFullAccess</code> 以及 <code>AliyunSTSAssumeRoleAccess</code> 的权限。具体操作请参见权限管理。

参数	说明
预设OSS路径	如果当前账号仅拥有某个Bucket或Bucket下某个路径的权限，需填写预设OSS路径。 预设OSS路径格式为 <code>oss://bucketname/path</code>。例如授权访问存储空间examplebucket下文件夹examplefolder下的文件或子文件夹，则填写 <code>oss://examplebucket/examplefolder/</code>。 当授权访问的Bucket开启了请求者付费模式，且您不是该Bucket拥有者的情况下，需选中请求者付费模式。否则，访问预设OSS路径下的指定资源时会报错 <code>AccessDenied</code>。选中请求者付费模式后可正常访问预设OSS路径下的指定资源，且访问该Bucket产生的流量、请求次数等费用将由您支付。有关请求者付费模式的更多信息，请参见请求者付费模式。
区域	当Endpoint配置为 默认（公共云） 时，需填写预设OSS路径对应Bucket所在的 区域 。
保持登录	选中后，ossbrowser会保持登录状态，下次打开时将自动登录。
记住密钥	选中可保存AK密钥。再次登录时，单击AK历史，可选择指定密钥直接登录。  警告 为避免不必要的信息安全风险，请勿在临时使用的电脑上选中该项。

■ 通过授权码登录

如果您想要授权其他人临时访问您Bucket下的指定资源，请选择授权码登录。授权码到期后会自动失效。使用授权码登录的具体步骤，请参见**使用临时授权码登录**。

3.2. 快速使用ossbrowser

ossbrowser是阿里云官方提供的OSS图形化管理工具，提供类似Windows资源管理器的功能。使用ossbrowser，您可以快速完成存储空间（Bucket）和文件（Object）的相关操作。

前提条件

已安装并登录ossbrowser。具体操作，请参见**安装并登录ossbrowser**。

Bucket或Object相关操作

ossbrowser支持的Bucket或Object级别的操作与控制台支持的操作类似，请按照ossbrowser界面指引完成Bucket或Object相关操作。

分类	配置说明
----	------

分类	配置说明
Bucket相关操作	- 创建Bucket Bucket是您用于存储Object的容器。在上传任何文件到OSS之前，您必须先创建存储空间。有关创建Bucket时如何填写Bucket名称、选择所在地域、ACL权限和存储类型信息，请参见创建存储空间。 - 删除Bucket 如果您不再需要Bucket，请将其删除，以免产生额外费用。有关删除Bucket的注意事项，请参见删除存储空间。
Object相关操作	ossbrowser支持上传、下载、预览、移动或复制文件、分享文件等操作。操作过程中有如下注意事项： - ossbrowser默认使用分片上传和断点续传上传文件，上传文件最大不能超过48.8 TB。若您因意外中断了文件上传的过程，且未继续完成该文件的上传，则已上传的部分会以碎片（Part）的形式存储在OSS的存储空间（Bucket）中。若您不再需要这些Part，建议您通过以下方式删除，以免产生额外的存储费用。 - 手动删除Part的具体操作，请参见删除碎片。 - 通过生命周期规则自动删除Part的具体操作，请参见设置生命周期规则。  注意 如果上传的文件与Bucket中已有的文件重名，则覆盖已有文件。 - 移动或复制文件最大不能超过5 GB，5 GB以上文件的移动或复制操作建议使用ossutil。 - 文件上传至Bucket后，您可以将文件URL分享给第三方，供其下载或预览。具体操作，请参见分享文件。 Object各类操作的其他注意事项，请参见控制台用户指南对应文档。

3.3. 权限管理

本文主要介绍如何通过ossbrowser进行简单的权限管理。

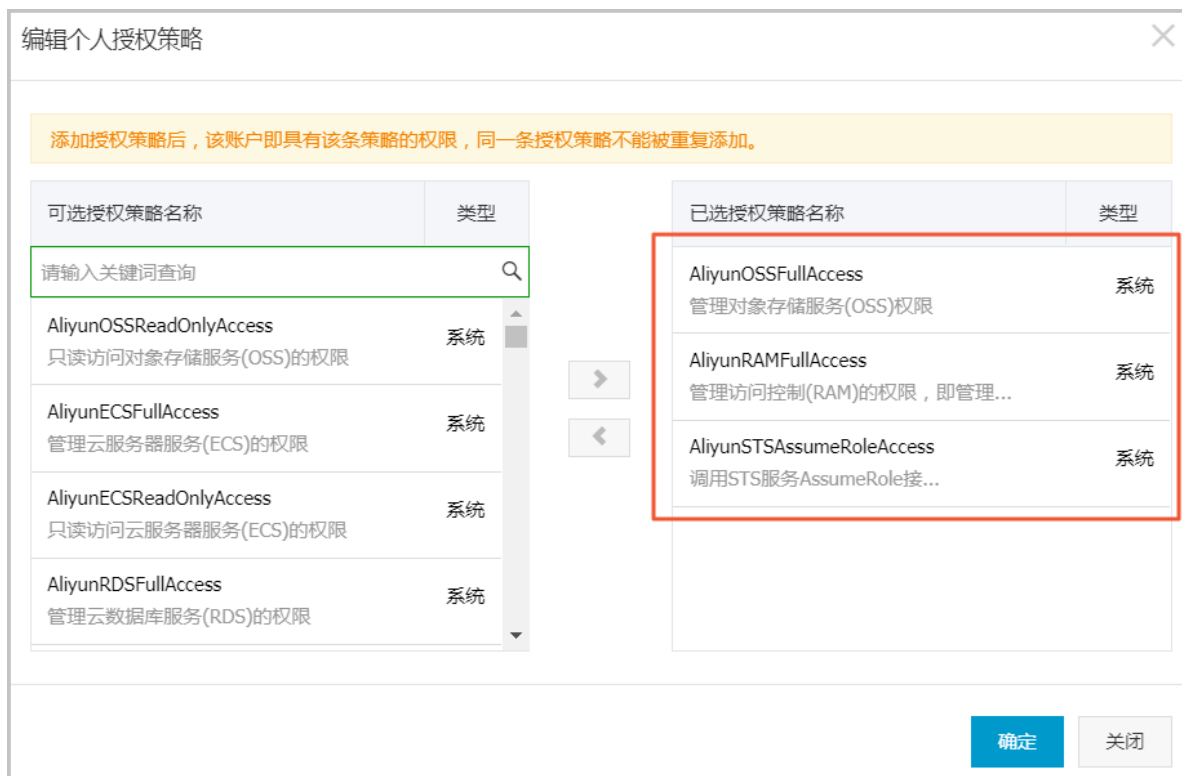
RAM用户

为保证数据安全，推荐您使用RAM用户（子账号）的AccessKey（AK）登录ossbrowser。

 **说明** 创建RAM用户和AK。详情请参见[创建RAM用户](#)。

RAM用户的权限分为：

- 管理员子账号：拥有管理权限的RAM用户。例如授予某个RAM用户可管理所有Bucket和具有RAM授权配置的权限，该RAM用户即为管理员子账号。您可以使用阿里云账号登录RAM控制台，创建管理员子账号，对账号授予如下权限。



- 操作员子账号：仅拥有某个Bucket或某个目录只读权限的RAM用户。管理员可通过[简化Policy授权](#)给RAM用户分配权限。

说明 您可以为RAM用户授予更小的权限，详情请参见[RAM Policy概述](#)。

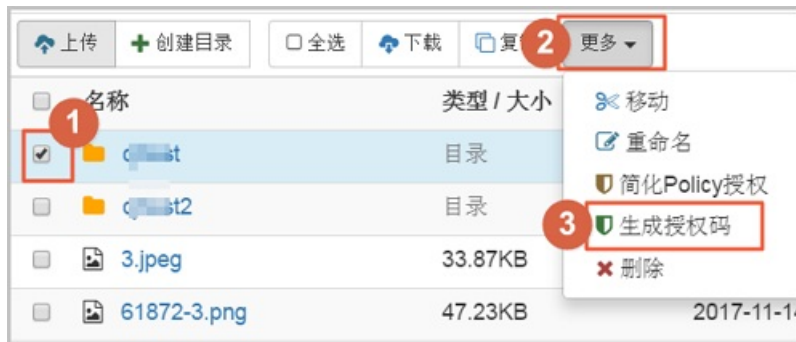
使用临时授权码登录

ossbrowser支持临时授权码登录。您可以将临时授权码提供给相应的人员，允许其在授权码到期前，临时访问您Bucket下某个目录。到期后，临时授权码会自动失效。

1. 使用管理员子账号登录ossbrowser。

注意 为了您的数据安全，当您使用阿里云账号或拥有所有管理权限的管理员子账号登录ossbrowser时，部分功能会被限制。您需使用管理员子账号的AK登录ossbrowser，生成授权码。管理员子账号需拥有对应存储空间或文件目录的管理权限、管理访问控制（RAM）的权限（AliyunRAMFullAccess）、调用STS服务AssumeRole接口的权限（AliyunSTSAssumeRoleAccess）。

2. 勾选需要临时授权的文件或目录，单击**更多 > 生成授权码**。



3. 将获取的授权码保存。
4. 退出登录，并使用授权码登录ossbrowser，如下图所示。



简化Policy授权

管理员子账号登录ossbrowser后，可通过**简化Policy授权**，创建操作员子账号或对操作员子账号进行授权，可授予某个Bucket或某个目录只读或读写权限。

② 说明 ossbrowser简化Policy授权，是基于阿里云RAM的访问控制产品。您也可以直接登录阿里云官网的RAM控制台，对RAM用户进行更细致的管理。

1. 使用管理员子账号登录ossbrowser。
2. 勾选一个或多个需要授权的文件或目录，并单击**更多 > 简化Policy授权**。
3. 在**简化Policy授权**页面，选择权限。
4. 您可以在该页面给已有的操作员子账号授权或者创建新的操作员子账号。



您可以查看生成的Policy文本，并将其复制到您需要的地方使用。例如，将Policy文本复制到阿里云官网RAM控制台，用于RAM用户、Role的授权策略编辑。

 **注意** 当前登录ossbrowser的AK，必须拥有RAM的配置操作权限，才能使用简化Policy授权。例如拥有RAM配置管理权限的管理员子账户的AK。

3.4. 常见问题

本文介绍在使用ossbrowser时遇到的常见问题及解决方法。

使用AccessKey登录时出现 “AccessDenied:You are forbidden to list buckets” 报错

问题分析：当前账号没有访问所有Bucket的权限。

解决方法：

- 为当前账号授予 `oss:ListBuckets` 权限后重新登录。具体步骤，请参见[为RAM用户授权](#)。
- 如果当前账号仅有部分Bucket或部分文件的访问权限，请在预设OSS路径栏添加访问路径，并选择Bucket所在区域。具体步骤，请参见[安装并登录ossbrowser](#)。

上传、下载文件较多时，如何提高上传、下载速度？

问题分析：ossbrowser默认同时执行3个上传或下载任务，剩余的任务会在队列中等待。在文件较多的情况下，默认设置会影响上传、下载的速度。

解决方法：在ossbrowser操作界面，单击**设置**，然后修改上传下载的任务数。您设备的硬件会影响这两个数值，建议您多次调整测试，以获取最佳数值。



macOS系统无法打开ossbrowser

通过以下步骤解决macOS系统无法打开ossbrowser的问题。

1. 解压已下载的安装包。
2. 将 `oss-browser.app` 拷贝至Application目录。

3. 右击Application目录下的oss-browser图标，选择Open。

Windows系统安装某软件后无法打开ossbrowser

问题分析：该软件为Windows系统的环境变量增加了 `NODE_OPTIONS=--max-http-header-size=value` 变量，导致ossbrowser无法打开。

解决方法：删除 `NODE_OPTIONS=--max-http-header-size=value` 变量。以Windows10 64位系统为例，步骤如下：

1. 进入系统页面，路径为控制面板\系统和安全\系统。
2. 单击高级系统设置。
3. 在系统属性对话框，单击环境变量。
4. 在用户变量列表选中NODE_OPTIONS=--max-http-header-size= value，然后单击删除。
5. 单击确定。

设置完成后即可正常打开ossbrowser。

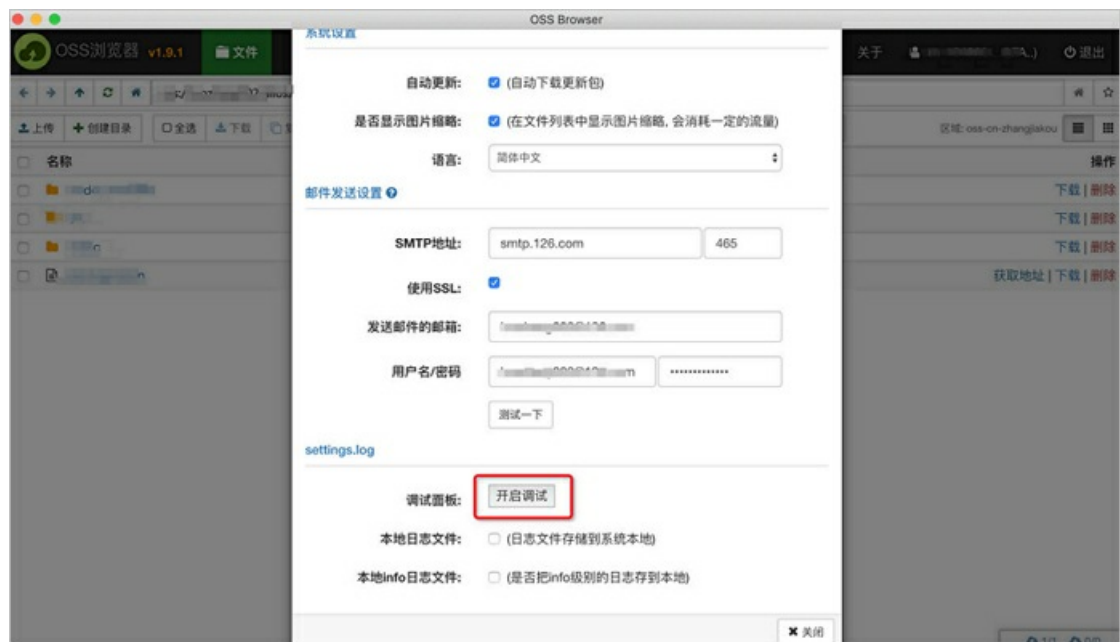
如何打开调试模式

您可以通过以下两种方式打开调试模式：

- 方式一：连续单击10次左上角应用Logo图标弹出调试面板。



- 方式二：单击设置页面的开启调试，弹出调试面板。



因网络原因无法正常访问ossbrowser怎么办？

可能原因是您的代理配置异常、或者存在无法连接互联网的代理配置。各操作系统查看代理配置的路径说明如下：

操作系统	查看代理配置路径
Windows	在Internet选项中查看代理配置。
macOS	在网络偏好设置中查看代理配置。
Linux	在系统设置 > 网络 > 网络代理中查看代理配置。

上传.asar文件提示“正在添加到上传队列”

原因是ossbrowser不支持上传.asar类型的文件。

4. 数据迁移工具ossimport

4.1. 说明及配置

ossimport是一款将数据迁移至OSS的工具。您可以将ossimport部署在本地服务器或云上ECS实例内，轻松将您本地或其它云存储的数据迁移到OSS。

支持特性

- 支持丰富的数据源，包括本地文件、七牛云Kodo、百度BOS、AWS S3、Azure Blob、又拍云USS、腾讯云COS、金山KS3、HTTP/HTTPS URL列表、阿里云OSS等，并可根据需要扩展。
- 支持单机模式和分布式模式。单机模式部署简单使用方便，分布式模式适合大规模数据迁移。
- 支持断点续传。
- 支持流量控制。
- 支持迁移指定时间以后的文件、特定前缀的文件。
- 支持并行数据下载和上传。

注意事项

- 您也可以使用[在线迁移服务](#)迁移您的数据，无需部署迁移工具。
- 对于迁移小于30 TB的数据，推荐使用[ossutil](#)。该工具简单方便，通过-u, --update和--snapshot-path选项可以实现文件的增量上传，详细说明请参见[cp](#)。

运行环境

ossimport可以部署在Linux或Windows系统上，要求如下：

- Windows7及以上版本
- Linux系统最新版本
- Java 1.7

 **注意** 分布式部署暂时不支持Windows系统。

部署方式选择

ossimport有单机模式和分布式模式两种部署方式。

- 单机模式：当您需要迁移的数据小于30 TB时，推荐部署[单机模式](#)。您可以将ossimport部署在任意一台可以访问您待迁移数据，且可以访问OSS的机器上。
- 分布式模式：当您需要迁移的数据大于30 TB时，推荐部署[分布式模式](#)。您可以将ossimport部署在任意多台可以访问您待迁移数据，且可以访问OSS的机器上。

 **说明** 当您待迁移的数据过大时，为了节约时间，您可以将ossimport部署到与您OSS相同地域的ECS实例上，并通过专线将源数据存放的服务器挂载到阿里云VPC网络中。多台ECS实例将数据通过内网迁移至OSS，会极大的提升数据迁移效率。

单机模式

Master、Worker、Tracker、Console运行在一个机器上，统一打包成 `ossimport2.jar`。系统中有且只有一个Worker。

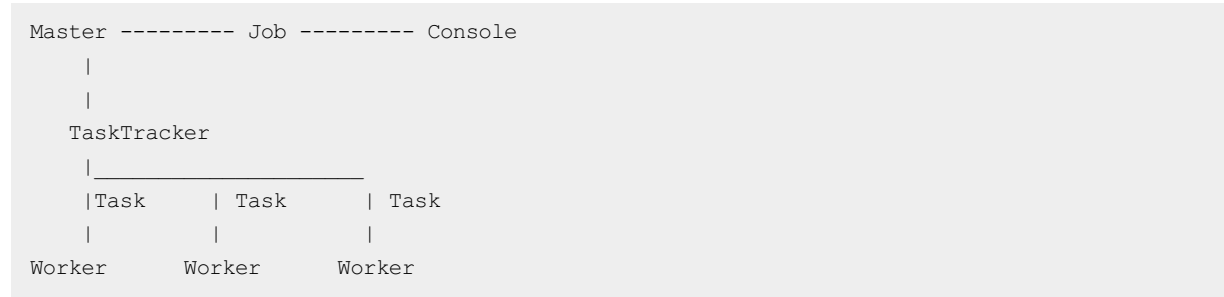
单机模式下文件结构如下：

```
ossimport
├── bin
│   └── ossimport2.jar # 包括Master、Worker、Tracker、Console四个模块的总jar
├── conf
│   ├── local_job.cfg # 单机Job配置文件
│   └── sys.properties # 系统运行参数配置文件
├── console.bat # Windows命令行，可以分布执行调入任务
├── console.sh # Linux命令行，可以分布执行调入任务
├── import.bat # Windows一键导入，执行配置文件为conf/local_job.cfg配置的数据迁移任务，包括启动、迁移、校验、重试
├── import.sh # Linux一键导入，执行配置文件为conf/local_job.cfg配置的数据迁移任务，包括启动、迁移、校验、重试
├── logs # 日志目录
└── README.md # 说明文档，强烈建议使用前仔细阅读
```

- import.bat/import.sh为一键导入脚本，修改完 local_job.cfg 后可以直接运行。
- console.bat/console.sh为命令行工具，可以用于分布执行命令。
- 脚本或命令请在 ossimport 目录下执行，即 *.bat/*.sh 的同级目录。

分布式模式

ossimport是基于master-worker的分布式架构，结构如下：



参数	说明
Master	负责将Job切分成Task，按照数据大小和文件个数分解成Task，数据大小/文件个数可以在sys.properties中配置。Job切分成Task的详细过程如下： 1. Master从本地/其它云存储中遍历出完整的待迁移的文件列表。 2. 按照数据大小和文件个数把完整的文件列表切分成Task，每个Task负责部分文件的迁移或校验。
Worker	• 负责Task的文件迁移和数据校验，从数据源上拉取指定文件，并上传到OSS的指定目录。迁移的数据源和OSS的配置在job.cfg或local_job.cfg中指定。 • Worker数据迁移支持限流、指定Task并发数，在sys.properties中配置。
TaskTracker	简称Tracker，负责Task的分发、Task状态跟踪。

参数	说明
Console	负责与用户交互，接受命令显示结果。支持系统管理命令 deploy/start/stop, Job管理命令 submit/retry/clean。
Job	用户提交的数据迁移任务，对用户来说一个任务对应一个配置文件 <code>job.cfg</code> 。
Task	Job按照数据大小和文件个数可以分成多个Task，每个Task 迁移部分文件。Job切分成Task的最小单位是文件，同一个文件不会切分到多个Task中。

分布式模式下可以启动多个Worker执行迁移数据，Task平均分配到Worker上执行，一个Worker执行多个Task。每一个机器上只能启动一个Worker。`workers` 配置的第一个Worker上会同时启动Master、Tracker。Console也要在该机器上运行。

分布式模式下文件结构如下：

ossimport	
├─ bin	
│ └─ console.jar	# Console模块jar包
│ └─ master.jar	# Master模块jar包
│ └─ tracker.jar	# Tracker模块jar包
│ └─ worker.jar	# Worker模块jar包
├─ conf	
│ └─ job.cfg	# Job配置文件模板
│ └─ sys.properties	# 系统运行参数配置文件
│ └─ workers	# Worker列表
├─ console.sh	# 命令行工具，目前支持只Linux
├─ logs	# 日志目录
└─ README.md	# 说明文档，强烈建议使用前仔细阅读

配置文件

单机模式下有两个配置文件`sys.properties`、`local_job.cfg`，分布式模式下有三个配置文件`sys.properties`、`job.cfg`、`workers`。其中`local_job.cfg`和`job.cfg`的配置项是完全一样的，只是名称不一样，`workers`是分布式环境独有的。

- `sys.properties`：系统运行参数

参数	含义	说明
workingDir	工作目录	工具包解压后的目录。单机模式下请不要修改此参数，分布式模式下每个机器的工作目录必须相同。
workerUser	Worker机器的ssh用户名	◦ 如果配置了privateKeyFile，则优先使用privateKeyFile。 ◦ 如果没有配置privateKeyFile，则使用workerUser/workerPassword。 ◦ 单机模式不需要修改此参数。
workerPassword	Worker机器的ssh用户密码	单机模式不需要修改此参数。

参数	含义	说明
privateKeyFile	private key文件路径	- 如果已经打通了ssh通道，则可以指定此参数，否则为空。 - 如果配置了privateKeyFile，则优先使用privateKeyFile。 - 如果没有配置privateKeyFile，则使用workerUser/workerPassword。 - 单机模式不需要修改此参数。
sshPort	ssh端口	默认22，一般情况无需更改。单机模式不需要修改此参数。
workerTaskThreadNum	Worker执行Task的最大线程数	- 该参数与机器的内存/网络有关，建议值60。 - 物理机可以适当加大，例如150。如果网络带宽已占满，请不要再加大。 - 如果网络较差，请适当降低，例如30。防止请求竞争网络造成大量请求超时。
workerMaxThroughput(KB/s)	worker数据迁移的流量上限	该值能起到限流作用，默认0表示不限流。
dispatcherThreadNum	Tracker的Task分发与状态确认的线程数	默认值一般够用，没有特殊需要请不要修改默认值。
workerAbortWhenUncatchedException	表示遇到未知错误时是跳过还是终止	默认跳过。
workerRecordMd5	在OSS中是否使用元数据x-oss-meta-md5记录迁移文件MD5值，默认不记录。	主要用于用户使用MD5校验文件数据。

- job.cfg：数据迁移任务配置，`local_job.cfg` 和 `job.cfg` 的配置项是完全一样的，只是名称不一样。

参数	含义	说明
jobName	任务名字，字符串。	- 任务的唯一标识，命名规则 [a-zA-Z0-9_-]{4,128}，支持提交多个名字不同的任务。 - 如果重复提交同名任务会提示任务已存在，清理（clean）同名任务前，无法提交同名任务。
jobType	任务类型，字符串	两类 <code>import</code> 或 <code>audit</code> ，默认 <code>import</code> 。 <code>import</code>，执行数据迁移操作并校验迁移数据的一致性。 <code>audit</code>，仅进行数据一致性校验。

参数	含义	说明
isIncremental	是否打开增量迁移模式，布尔类型。	- 默认值false。 - 如果设为true，会每间隔 incrementalModelInterval（单位秒）重新扫描一次增量数据，并将增量数据迁移到OSS。
incrementalModelInterval	增量模式下的同步间隔，整型，单位秒。	isIncremental=true时有效。可配置的最小间隔为900秒，不建议配置成小于3600秒的值，会浪费大量请求，造成额外的系统开销。
importSince	迁移大于该时间的数据，整型，单位秒。	- 该时间为 Unix时间戳，即自1970年1月1日UTC零点以来的秒数，通过命令date +%s获取。 - 默认为0，表示迁移全部数据。
srcType	同步源类型，字符串，请注意大小写。	支持以下类型： <code>local</code>：从本地文件迁移数据到OSS，该选项只需要填写srcPrefix，不需要填写srcAccessKey, srcSecretKey, srcDomain, srcBucket。 <code>oss</code>：从一个 OSS bucket 迁移到另一个 bucket。 <code>qiniu</code>：从七牛云存储迁移到OSS。 <code>bos</code>：从百度的云存储迁移到OSS。 <code>ks3</code>：从金山云存储迁移到OSS。 <code>s3</code>：从 AWS S3 迁移到OSS。 <code>youpai</code>：从又拍云迁移到OSS。 <code>http</code>：通过提供的HTTP或者HTTPS链接列表迁移数据到OSS。 <code>cos</code>：从腾讯云存储COS迁移到OSS。 <code>azure</code>：从Azuer Blob迁移到OSS。
srcAccessKey	源AccessKey，字符串。	- 如果srcType设置为 <code>oss</code>、<code>qiniu</code>、<code>baidu</code>、<code>ks3</code>、<code>s3</code>，则填写数据源的AccessKey。 - 如果srcType设置为 <code>local</code>、<code>http</code>，则该项不需要填写。 - 如果srcType设置为 <code>youpai</code>、<code>azure</code>，则填写用户名AccountName。

参数	含义	说明
srcSecretKey	源SecretKey, 字符串。	- 如果srcType设置为 <code>oss</code>、<code>qiniu</code>、<code>baidu</code>、<code>ks3</code>、<code>s3</code>，则填写数据源的SecretKey。 - 如果srcType设置为 <code>local</code>、<code>http</code>，则该项不需要填写。 - 如果srcType设置为 <code>youpai</code>，则填写操作员密码。 - 如果srcType设置为 <code>azure</code>，则填写AccountKey。
srcDomain	源Endpoint	- 如果srcType设置为 <code>local</code>、<code>http</code>，则该项不需要填写。 - 如果srcType设置为 <code>oss</code>，则填写从控制台获取的域名，非带bucket前缀的二级域名。 - 如果srcType设置为 <code>qiniu</code>，则填写从七牛控制台获取的对应bucket的域名。 - 如果srcType设置为 <code>bos</code>，则填写百度BOS域名，如 <code>http://bj.bcebos.com</code> 或 <code>http://gz.bcebos.com</code>。 - 如果srcType设置为 <code>ks3</code>，则填写金山ks3域名，如 <code>http://kss.ksyun.com</code> 或 <code>http://ks3-cn-beijing.ksyun.com</code> 或 <code>http://ks3-us-west-1.ksyun.com</code>。 - 如果srcType设置为 <code>s3</code>，则填写AWS S3各region的域名。 - 如果srcType设置为 <code>youpai</code>，则填写又拍云域名，如自动判断最优线路 <code>http://v0.api.upyun.com</code> 或电信线路 <code>http://v1.api.upyun.com</code> 或联通网通线路 <code>http://v2.api.upyun.com</code> 或移动铁通线路 <code>http://v3.api.upyun.com</code>。 - 如果srcType设置为 <code>cos</code>，则填写腾讯云bucket所在的区域，例如华南园区为 <code>ap-guangzhou</code>。 - 如果srcType设置为 <code>azure</code>，则填写Azure Blob连接字符串中的EndpointSuffix，如 <code>core.chinacloudapi.cn</code>。
srcBucket	源bucket名字或container名称	- 如果srcType设置为 <code>local</code>、<code>http</code>，则不需要填写。 - 如果srcType设置为 <code>azure</code>，则填写container名称。 - 其它填写bucket名称。

参数	含义	说明
srcPrefix	源前缀，字符串，默认为空	- 如果srcType设置为local，则填写本地目录，需要完整路径，以单个正斜线（/）进行分割并且以单个正斜线（/）结尾，仅支持如 <code>c:/example/</code> 或者 <code>/data/example/</code> 的格式。  注意 <code>c:/example//</code> 或 <code>/data//example/</code> 或 <code>/data/example//</code> 是非法的。 - 如果srcType设置为 <code>oss</code>、<code>qiniu</code>、<code>bos</code>、<code>ks3</code>、<code>youdao</code>、<code>s3</code>，则填写待同步object的前缀，不包括bucket名称，如 <code>data/to/oss/</code>。 - 如需同步所有文件，则srcPrefix设置为空。
destAccessKey	目的AccessKey，字符串。	访问OSS服务的AccessKey ID，请到 阿里云控制台 查看。
destSecretKey	目的SecretKey，字符串。	访问OSS的AccessKey Secret，请到 阿里云控制台 查看。
destDomain	目的endpoint，字符串。	从 阿里云控制台 获取，非带bucket前缀的二级域名，列表请参看域名列表。
destBucket	目的bucket，字符串。	OSS的bucket名称，不需要以/结尾。
destPrefix	目标前缀，字符串，默认为空。	- 目标前缀，默认为空，直接放在目标bucket下。 - 如果要将数据同步到 <code>oss</code> 的某个目录下，请以/结尾，如 <code>data/in/oss/</code>。 - 注意 <code>oss</code> 不支持以/作为文件的开头，所以destPrefix请不要配置以/做为开头。 - 一个本地文件路径为<code>srcPrefix+relativePath</code>的文件，迁移到 <code>oss</code> 的路径为<code>destDomain/destBucket/destPrefix+relativePath</code>。 - 一个云端文件路径为<code>srcDomain/srcBucket/srcPrefix+relativePath</code>的文件，迁移到 <code>oss</code> 的路径为<code>destDomain/destBucket/destPrefix+relativePath</code>。

参数	含义	说明
taskObjectCountLimit	每个Task最大的文件数，整型，默认10000。	该配置项会影响到任务执行的并行度，一般配置为总文件数/Worker总数/迁移线程数（workerTaskThreadNum），最大值不要超过50000，如果不知道总文件数，请使用默认值。
taskObjectSizeLimit	每个Task最大数据量，整型，单位bytes，默认1GB。	该配置项会影响到任务执行的并行度，一般配置为总数据量/Worker总数/迁移线程数（workerTaskThreadNum），如果不知道总数据量，请使用默认值。
isSkipExistFile	数据迁移时是否跳过已经存在的文件，布尔类型。	当设置为true时，根据文件的size和LastModifiedTime判断是否跳过；为false时，总是覆盖OSS上已有文件。默认为值为false。jobType为audit时此项不生效。
scanThreadCount	并行扫描文件的线程数，整型。 ◦ 默认值：1 ◦ 有效值：1-32	该配置项与扫描文件的效率有关，没有特殊需求请不要修改。
maxMultiThreadScanDepth	最大允许并行扫描目录的深度，整型。 ◦ 默认值：1 ◦ 有效值：1-16	◦ 默认值1表示在顶级目录间并行扫描。 ◦ 没有特殊需求不要修改，随意配置过大值会导致任务无法正常运行。
appId	腾讯云COS的appId，整型。	srcType=coss时有效。
httpListFilePath	HTTP列表文件的绝对路径，字符串。	◦ srcType=http时有效，源为HTTP链接地址时，需要提供内容为HTTP链接地址文件的绝对路径，如c:/example/http.list。 ◦ 该文件中的HTTP链接需要划分成两列，以空格分开，分别代表前缀和上传到OSS后的相对路径。例如c:/example/http.list文件内容如： http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/aa/bb.jpg http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/cc/dd.jpg 两行，迁移到OSS的文件名分别是 destPrefix + bb.jpg和 destPrefix + cc/dd.jpg。

- workers：分布式模式独有，每个IP一行。如：

```
192.168.1.6
192.168.1.7
192.168.1.8
```

- 上述配置情况下，第一行的 192.168.1.6 一定是 master；即 192.168.1.6 上会同时启动Master、Worker、TaskTracker。Console也需要在该机上运行。
- 多个Worker机器的用户名、登录方式、工作目录请确保相同。

配置文件示例

下表中是分布式部署下的数据迁移任务配置文件，单机的配置文件名是 `local_job.cfg`，配置项与分布式部署时没有区别。

迁移类型	配置文件	说明
从本地迁移到OSS	<code>job.cfg</code>	<code>srcPrefix</code> 是以正斜线（/）结尾的绝对路径，如 <code>D:/work/oss/data/</code> 、 <code>/home/user/work/oss/data/</code> 。
从七牛云存储迁移到OSS	<code>job.cfg</code>	<code>srcPrefix</code> 和 <code>destPrefix</code> 可以配置为空；如果不为空，请以正斜线（/）结尾，如 <code>destPrefix=docs/</code> 。
从百度bos迁移到OSS	<code>job.cfg</code>	<code>srcPrefix</code> 和 <code>destPrefix</code> 可以配置为空；如果不为空，请以正斜线（/）结尾，如 <code>destPrefix=docs/</code> 。
从AWS S3迁移到OSS	<code>job.cfg</code>	参见 S3域名列表 。
从又拍云存储迁移到OSS	<code>job.cfg</code>	<code>srcAccessKey</code> / <code>srcSecretKey</code> 填操作员账号及密码。
从腾讯cos迁移到OSS	<code>job.cfg</code>	<code>srcDomain</code> 请按照V4版本填写，如 <code>srcDomain=sh</code> 。 <code>srcPrefix</code> 可以为空，当不为空时候，请以 <code>/</code> 开头和结尾，如 <code>srcPrefix=/docs/</code> 。
从Azure blob迁移到OSS	<code>job.cfg</code>	<code>srcAccessKey</code> / <code>srcSecretKey</code> 填写存储账号及密钥。 <code>srcDomain</code> 填连接字符串中的 <code>EndpointSuffix</code> ，如 <code>core.chinacloudapi.cn</code> 。
从OSS迁移到OSS	<code>job.cfg</code>	适用于不同区域之间、不同存储类型之间、不同前缀之间的数据迁移。推荐在ECS上部署，并使用带internal的域名，可以节省流量费用。

高级设置

- 分时限流

`sys.properties`中的`workerMaxThroughput`(KB/s)表示Worker流量的上限，如果业务需要限流，例如源端流控控制、网络限制等情况。该参数的值应该小于机器的最大网络流量，并根据业务需要评估。修改后需要重启服务才能生效。

分布式部署情况下，需要修改每个Worker的`$OSS_IMPORT_WORK_DIR/conf`下的`sys.properties`，然后重启服务。

要实现分时限流，可通过`crontab`定时修改`sys.properties`，然后重启服务生效。
- 修改任务并发数
 - `sys.properties`中的`workerTaskThreadNum`表示Worker并发执行的任务数量，如果网络较差、并发大，会出现大量超时错误，此时应该修改此配置，降低并发量，并重启服务。
 - `sys.properties`中的`workerMaxThroughput`(KB/s)表示Worker流量的上限，如果业务需要限流，例如源端流控控制、网络限制等情况。该参数的值应该小于机器的最大网络流量，并根据业务需要评估。
 - `job.cfg`中的`taskObjectCountLimit`，每个Task最大的文件数，默认10000。该参数会影响Task的数量，数量过小无法实现有效的并发。

- `job.cfg`中的`taskObjectSizeLimit`，每个Task最大数据量，默认1GB。该参数会影响Task的数量，数量过小无法实现有效的并发。

注意

- 请在启动迁移任务前，完成各项参数配置。
- `sys.properties`中的参数修改后，重启迁移服务器后才能生效。
- `job.cfg`任务提交后，任务的配置参数无法更改。

只校验不迁移数据

ossimport支持只校验数据不迁移数据，设置任务配置文件`job.cfg`或`local_job.cfg`的配置项`jobType`为`audit`而不是`import`，其它配置与数据迁移相同。

数据迁移增量模式

数据迁移增量模式，是指数据迁移任务启动后，先进行一次全量迁移，每隔一段时间自动的进行增量数据迁移。第一次数据迁移任务为全量迁移，提交任务后立即启动；后面的增量数据迁移每隔一个周期启动一次。数据迁移增量模式适用于数据备份和数据同步。

增量模式有两个配置项：

- `job.cfg`中的`isIncremental`，表示是否打开增量迁移模式，`true`表示打开增量模式，`false`表示关闭增量模式，默认关闭。
- `job.cfg`中的`incrementalModelInterval`，表示增量模式下的同步间隔，即增量数据迁移的间隔周期，单位秒。`isIncremental=true`时有效。可配置的最小值为900秒，不建议配置成小于3600秒的值，会浪费大量请求，造成额外的系统开销。

指定迁移文件的过滤条件

迁移文件的过滤条件，即只迁移满足特定条件的文件。ossimport支持指定前缀和最后修改时间：

- `job.cfg`中的`srcPrefix`，用来指定迁移文件的前缀，默认为空。
 - 如果`srcType=local`，则填写本地目录。填写时需要完整路径，以正斜线（/）进行分割并且以正斜线（/）结尾，如 `c:/example/` 或 `/data/example/`。
 - 如果`srcType`为`oss`、`qiniu`、`bos`、`ks3`、`youpai`、`s3`，则为待同步object的前缀，不包括bucket名称。如 `data/to/oss/`。迁移所有文件时`srcPrefix`设置为空。
- `job.cfg`中的`importSince`，用来指定迁移文件的最后修改时间，单位秒。`importSince`为Unix时间戳，即自1970年1月1日UTC零点以来的秒数，通过命令`date +%s`获取。默认值0，表示迁移全部数据。增量模式下只对第一次全量迁移有效，非增量模式对整个迁移任务有效。
 - 如果文件的`LastModified Time`在`importSince`之前，文件不被迁移。
 - 如果文件的`LastModified Time`在`importSince`之后，文件将被迁移。

4.2. 单机部署

本文介绍如何单机部署ossimport，单机部署支持Linux和Windows系统。

背景信息

已安装Java 1.7及1.8版本。

快速使用

1. 下载ossimport-2.3.5.zip并解压。

解压后的文件结构如下：

```
ossimport
├── bin
│   └── ossimport2.jar # 包括Master、Worker、TaskTracker、Console四个模块的总jar
├── conf
│   ├── local_job.cfg # Job配置文件
│   └── sys.properties # 系统运行参数配置文件
├── console.bat # Windows命令行，可以分布执行调入任务
├── console.sh # Linux命令行，可以分布执行调入任务
├── import.bat # Windows一键导入，执行配置文件为conf/local_job.cfg配置的数据迁移任务，包括启动、迁移、校验、重试
├── import.sh # Linux一键导入，执行配置文件为conf/local_job.cfg配置的数据迁移任务，包括启动、迁移、校验、重试
├── logs # 日志目录
└── README.md # 说明文档，强烈建议使用前仔细阅读
```

2. 根据需求编辑conf/sys.properties、conf/local_job.cfg文件。

注意不要修改以下内容：

- conf/sys.properties中的配置项 `workingDir`、`workerUserName`、`workerPassword`、`privateKeyFile`。
- conf/local_job.cfg 的名称、位置、配置项 `jobName`。

配置示例请参见[配置文件示例](#)。

 **说明** 请在提交任务前确认 `sys.properties` 和 `local_job.cfg` 中的参数，任务提交后参数无法再修改。

3. 配置完成后运行任务。

- Window系统下在双击运行 `import.bat`。
- Linux终端中执行 `bash import.sh` 命令。

运行方式

单机模式下，数据迁移任务有以下两种执行方式：

- 一键导入：是对所有步骤的封装，按照脚本提示执行即可完成数据迁移。[快速使用](#)步骤使用的是这种执行方式，对于初级用户建议使用一键导入。

i. 执行一键导入。

- Window系统下在双击运行 `import.bat`。
- Linux终端中执行 `bash import.sh` 命令。

 **说明** 如果之前执行过程序，会提示是否从上次的断点处继续执行，或者重新执行同步任务。对新的数据迁移任务，或者修改了同步的源端/目的端，请选择重新执行。

- ii. Windows下任务开始后，会打开一个新的cmd窗口执行同步任务并显示日志，旧窗口会每隔10秒打印一次任务状态，数据迁移期间不要关闭两个窗口；Linux下服务在后台执行。
- iii. 当Job完成后，如果发现任务失败了，会提示是否重试。输入 `y` 重试，输入 `n` 则跳过退出。
- iv. 如果上传失败，请查看 `master/jobs/local_test/failed_tasks/<taskaskid>/audit.log` 文件，确定失

败原因。

- 分步执行：执行启动服务、提交任务、重试失败子任务等步骤。

- i. 清除同名任务。

如果以前运行过同名任务，需要重新执行任务，请先清除同名任务。如果没有运行过，或需要重试失败任务，不要执行清除命令。

- Window下在cmd.exe中执行**console.bat clean**。
- Linux下在终端执行**bash console.sh clean**。

- ii. 提交数据迁移任务。

ossimport 不能提交同名任务，如果有请先清除。提交任务的配置文件为 *conf/local_job.cfg*，默认任务名称为 `local_test`。提交任务的命令：

- Window下在cmd.exe中执行**console.bat submit**。
- Linux下在终端执行**bash console.sh submit**。

- iii. 启动服务。

- Windows下在cmd.exe中执行**console.bat start**。
- Linux下在终端执行**bash console.sh start**。

- iv. 查看任务状态。

- Windows下在cmd.exe中执行**console.bat stat**。
- Linux下在终端执行**bash console.sh stat**。

- v. 失败Task重试。

由于网络或其它原因，子任务可能失败。失败重试只重试失败的Task，不会重试成功的Task。

- Windows下在cmd.exe中执行**console.bat retry**。
- Linux下在终端执行**bash console.sh retry**。

- vi. 停止服务。

- Windows下关闭窗口 `%JAVA_HOME%/bin/java.exe`。
- Linux下在终端执行**bash console.sh stop**。

任务状态及日志

任务提交后，Master分解成Task，由Worker执行Task，Tracker收集Task状态。任务运行完成后ossimport目录内容如下：

```
ossimport
├── bin
│   └── ossimport2.jar    # 单机版jar
├── conf
│   ├── local_job.cfg    # Job配置文件
│   └── sys.properties   # 系统运行参数配置文件
├── console.sh           # 命令行工具
├── import.sh            # 一键导入脚本
├── logs
│   ├── import.log       # 迁移日志
│   ├── job_stat.log     # 任务状态记录
│   ├── ossimport2.log   # 单机版运行日志
│   └── submit.log       # 任务提交记录
├── master
│   ├── jobqueue          # 存放尚未分解完成的任务
│   └── jobs              # 存放任务运行状态
│       ├── local_test   # 任务名称
│       │   ├── checkpoints # Master分解Job到Task的checkpoint点记录
│       │   │   └── 0
│       │   │       └── 034DC9DD2860B0CFE884242BC6FF92E7.cpt
│       │   ├── dispatched # 已经分配给Worker尚未运行完成的Task
│       │   │   └── localhost
│       │   ├── failed_tasks # 运行失败的Task
│       │   ├── pending_tasks # 尚未分配的Task
│       │   └── succeed_tasks # 成功运行的Task
│       │       └── A41506C07BF1DF2A3EDB4CE31756B93F_1499744514501@localhost
│       │           ├── audit.log # Task运行日志，通过该日志可以查看错误原因
│       │           ├── DONE      # Task运行成功标志
│       │           ├── error.list # Task错误列表，可以查看错误文件列表
│       │           └── STATUS    # 任务状态标志文件，内容为Failed或Completed，表示子任务失败或成功
│       └── TASK            # Task描述信息
└── worker                # Worker正在运行的Task状态，运行完成后由Master管理
    └── jobs
        ├── local_test
        └── tasks
```

 注意

- Job运行信息，可以查看 `logs/ossimport2.log`或`logs/import.log`。
- Task的失败原因，可以查看 `master/jobs/${JobName}/failed_tasks/${TaskName}/audit.log`。
- Task的失败文件，可以查看 `master/jobs/${JobName}/failed_tasks/${TaskName}/error.list`。
- 上述日志仅作为排查问题时的参考，您的业务和应用不要依赖于该内容。

迁移失败常见原因

- 上传过程中源目录的文件发生了修改，`log/audit.log`里会提示 `SIZE_NOT_MATCH` 相关字样的错误，这种情况下老的文件已经上传成功，新的修改没有上传到OSS。
- 源文件在上传过程中被删除，导致迁移失败。
- 源文件名不符合OSS命名规范（不能以正斜线（/）开头，不能为空），导致上传到OSS失败。

- 因网络异常、账号权限不足等原因导致的下载数据源文件失败，可查看 `logs/ossimport2.log` 或 `logs/import.log` 确定错误原因。
- 程序异常退出，任务状态为 Abort。如果遇到此类问题，请联系[技术支持](#)进行排查。

4.3. 分布式部署

本文介绍如何分布式部署ossimport，分布式部署目前只支持Linux系统，Windows系统暂不支持。

下载并安装ossimport

1. 下载ossimport

分布式版本下载地址[ossimport-2.3.5.tar.gz](#)，将ossimport安装包下载到本地。

2. 安装ossimport

- i. 登录服务器后执行以下命令，创建ossimport目录。

```
mkdir -p $HOME/ossimport
```

- ii. 进入压缩包当前目录，执行以下命令，将压缩包解压到指定目录。

```
tar -zxvf ossimport-2.3.5.tar.gz -C $HOME/ossimport
```

解压后的文件结构如下：

```
ossimport
├── bin
│   ├── console.jar    # Console模块jar包
│   ├── master.jar     # Master模块jar包
│   ├── tracker.jar    # Tracker模块jar包
│   └── worker.jar     # Worker模块jar包
├── conf
│   ├── job.cfg        # Job配置文件模板
│   ├── sys.properties # 系统运行参数配置文件
│   └── workers        # Worker列表
├── console.sh         # 命令行工具，目前支持只Linux
├── logs               # 日志目录
└── README.md         # 说明文档，使用前请仔细阅读
```

- `OSS_IMPORT_HOME`: ossimport的根目录，默认为解压命令中的目录 `$HOME/ossimport`，也可以通过命令 `export OSS_IMPORT_HOME=<dir>` 或修改系统配置文件 `$HOME/.bashrc` 设置，推荐使用默认目录；
- `OSS_IMPORT_WORK_DIR`: ossimport的工作目录，通过 `conf/sys.properties` 的配置项 `workingDir` 指定，推荐为 `$HOME/ossimport/workdir`；
- `OSS_IMPORT_HOME`或`OSS_IMPORT_WORK_DIR`请使用绝对路径配置，如 `/home/<user>/ossimport` 或 `/home/<user>/ossimport/workdir`。

配置

分布式部署有三个配置文件 `conf/sys.properties`、`conf/job.cfg`、`conf/workers`。

- `conf/job.cfg`：分布式模式下任务的配置文件模板，数据迁移前请按照实际参数修改。
- `conf/sys.properties`：系统运行参数配置文件，如工作目录、Worker运行参数等请在该文件中配置。

- `conf/workers` : worker列表。

注意

- 请在提交任务前确认 `sys.properties` 和 `job.cfg` 中的参数，任务提交后参数无法再修改。
- Worker列表 `workers` 请启动服务前确定，启动后无法再增加或删除。

运行

- 执行迁移任务

分布式部署时，执行任务的一般步骤是修改任务配置文件、部署服务、清除同名任务、提交任务、启动迁移服务、查看任务状态、重试失败子任务以及停止迁移任务。详细说明如下：

- 部署服务。在Linux终端执行 `bash console.sh deploy`。

 说明 部署前请保证配置文件 `conf/job.cfg`、`conf/workers` 已经修改完成。

- 清除同名任务。如果运行过同名任务，需要重新执行任务，请先清除同名任务。如果没有运行过，或需要重试失败子任务，不需要执行清除命令。Linux终端中执行 `bash console.sh clean job_name`。
- 提交数据迁移任务。OssImport不能重复提交同名任务，如果有请使用 `clean` 命令清除。提交任务需要指定任务的配置文件，任务的配置文件模板在 `conf/job.cfg`，建议在模板的基础上修改。在Linux终端执行 `bash console.sh submit [job_cfg_file]`，提交配置文件为 `job_cfg_file` 的任务，`job_cfg_file` 为可选参数，不指定时默认为 `$OSS_IMPORT_HOME/conf/job.cfg`，`$OSS_IMPORT_HOME` 默认为 `console.sh` 所在的目录。
- 启动服务。Linux终端执行 `bash console.sh start`。
- 查看任务状态。Linux终端执行 `bash console.sh stat`。
- 失败任务重试。由于网络或其它原因，任务可能运行失败。失败重试只重试失败的任务，成功的任务不会重试。Linux下在终端执行 `bash console.sh retry [job_name]`，`job_name` 为可选参数，指定 `job_name` 时重试失败任务的子任务，不指定 `job_name` 时重试所有任务的失败子任务。
- 停止服务。Linux终端执行 `bash console.sh stop`。

提示：

- `bash console.sh` 在参数错误时，会自动提示命令格式。
- 配置文件和提交任务中的目录，推荐使用绝对目录。
- 任务的配置，即 `job.cfg` 中的配置项。

 注意 配置项提交后不允许修改。

- 常见任务失败原因

- 上传过程中源目录的文件发生了修改，`log/audit.log` 里会提示 `SIZE_NOT_MATCH` 相关字样的错误，这种情况下老的文件已经上传成功，新的修改没有上传到OSS；
- 源文件在上传过程中被删除，导致下载的时候失败；
- 源文件名不符合OSS命名规范（不能以/开头，不能为空），导致上传到OSS失败；
- 下载数据源文件失败；
- 程序异常退出，任务状态为Abort，这种情况请联系我们。

● 任务状态及日志

任务提交后，Master分解成Task，有Worker执行Task，Tracker收集Task状态。任务运行完成后workdir目录内容如下：

```
workdir
├── bin
│   ├── console.jar    # Console模块jar包
│   ├── master.jar     # Master模块jar包
│   ├── tracker.jar    # Tracker模块jar包
│   └── worker.jar     # Worker模块jar包
├── conf
│   ├── job.cfg        # Job配置文件模板
│   ├── sys.properties # 系统运行参数配置文件
│   └── workers        # Worker列表
├── logs
│   ├── import.log     # 迁移日志
│   ├── master.log     # Master日志
│   ├── tracker.log    # Tracker日志
│   └── worker.log     # Worker日志
├── master
│   ├── jobqueue       # 存放尚未分解完成的任务
│   └── jobs           # 存放任务运行状态
│       └── xxtooss    # 任务名称
│           ├── checkpoints # Master分解Job到Task的checkpoint点记录
│           │   └── 0
│           │       └── ED09636A6EA24A292460866AFDD7A89A.cpt
│           ├── dispatched # 已经分配给Worker尚未运行完成的Task
│           │   └── 192.168.1.6
│           ├── failed_tasks # 运行失败的Task
│           │   └── A41506C07BF1DF2A3EDB4CE31756B93F_1499348973217@192.168.1.6
│           │       ├── audit.log # Task运行日志，通过该日志可以查看错误原因
│           │       ├── DONE      # Task运行成功标志，失败为空
│           │       ├── error.list # Task错误列表，可以查看错误文件列表
│           │       └── STATUS    # 任务状态标志文件，内容为Failed或Completed，表示子任务
失败或成功
│           │   └── TASK        # Task描述信息
│           ├── pending_tasks # 尚未分配的Task
│           └── succeed_tasks # 成功运行的Task
│               └── A41506C07BF1DF2A3EDB4CE31756B93F_1499668462358@192.168.1.6
│                   ├── audit.log # Task运行日志，通过该日志可以查看错误原因
│                   ├── DONE      # Task运行成功标志
│                   ├── error.list # Task错误列表，成功为空
│                   └── STATUS    # 任务状态标志文件，内容为Failed或Completed，表示子任务失
败或成功
│               └── TASK        # Task描述信息
└── worker # Worker正在运行的Task状态，运行完成后由Master管理
    └── jobs
        ├── local_test2
        │   └── tasks
        └── local_test_4
            └── tasks
```


 注意

- Job运行信息，可以查看 `logs/import.log`。
- Task的失败原因，可以查看 `master/jobs/${JobName}/failed_tasks/${TaskName}/audit.log`。
- Task的失败文件，可以查看 `master/jobs/${JobName}/failed_tasks/${TaskName}/error.list`。
- 上述日志仅作为排查问题时的参考，您的业务和应用不要依赖于该内容。

常见错误及排除

请查看[常见错误及排除](#)。

4.4. 常见问题

本文介绍使用数据迁移工具ossimport的常见问题及解决方法。

通用说明

本文提到的所有涉及ossimport的命令均为简写，实际操作时，需补全命令。

- Windows系统需加上`console.bat`进行补全，例如`submit`实际为`console.bat submit`。
- Linux系统需要加上`bash console.sh`进行补全，例如`submit`实际为`bash console.sh submit`。

迁移失败常见问题

如果迁移文件失败，建议您先查看迁移失败日志，确认失败的原因。您可以在解决这些问题后使用`retry`命令进行重试。迁移失败日志的路径为`master/jobs/${JobName}/failed_tasks/${TaskName}/audit.log`。

使用stat命令查看任务状态显示失败

```
[root@l-other-server3.bit.prod.aws.dm import]# bash console.sh stat
===== Process Status =====
[2019-04-26 08:49:47 0026][JobTracker][ INFO ] - [2019-04-26 08:49:47] [JobTracker] [INFO ] -
[2019-04-26 08:49:47] [JobTracker] [INFO ] - jobName:bi-s3-20190419
[2019-04-26 08:49:47] [JobTracker] [INFO ] - JobState:RUNNING
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Pending Task Count:0
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Dispatched Task Count:1624
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Succeed Task Count:787
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Failed Task Count:80
[2019-04-26 08:49:47] [JobTracker] [INFO ] - Is Scan Finished:true
[2019-04-26 08:49:47] [JobTracker] [INFO ] - =====
```

解决方法：使用`stat`命令查看迁移任务状态，如果`JobState`为`failed`，则迁移任务失败。迁移完成后请使用`retry`命令进行重试。

部分文件迁移失败后反复重试都无法成功迁移

解决方法：

1. 查看迁移失败文件列表`master/jobs/${JobName}/failed_tasks/${TaskName}/error.list`，获取失败文件的相对路径。
2. 确认是否有这部分文件的权限访问、文件是否被删除、是否是软链接文件、文件名是否存在乱码等。

3. 解决以上问题后，使用`retry`命令进行重试。

迁移失败日志报错 `The bucket you are attempting to access must be addressed using the specified endpoint.`

```
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.
<Error>
  <Code>AccessDenied</Code>
  <Message>The bucket you are attempting to access must be addressed using the specified endpoint. Please send all future requests to this endpoint.</Message>
  <RequestId>56EA98DE815804**21B23EE6</RequestId>
  <HostId>my-oss-bucket.oss-cn-qingdao.aliyuncs.com</HostId>
  <Bucket>my-oss-bucket</Bucket>
  <Endpoint>oss-cn-hangzhou.aliyuncs.com</Endpoint>
</Error>
```

- 问题原因：srcDomain或destDomain填写错误。
- 解决方法：请参见[访问域名和数据中心](#)填写正确的Endpoint。

迁移失败日志报错 `The request signature we calculated does not match the signature you provided.`

```
Exception:com.aliyun.oss.OSSException: The request signature we calculated does not match the signature you provided. Check your key and signing method.
[ErrorCode]: SignatureDoesNotMatch
[RequestId]: xxxxxxxx
[HostId]: xxx.oss-cn-shanghai.aliyuncs.com
```

- 问题原因：destAccessKey和destSecretKey填写错误。
- 解决方法：请填写正确的AccessKey信息。

迁移失败日志报错 `The bucket name "xxx/xx" is invalid.`

```
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.
```

- 问题原因：destBucket填写错误。
- 解决方法：填写正确的Bucket名称。关于Bucket名称的命名规范，请参见[存储空间命名](#)。

迁移失败日志报错 `Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 timed out.`

```
Unable to execute HTTP request: Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 timed out
[ErrorCode]: ConnectionTimeout
[RequestId]: Unknown
```

- 问题原因：这个是连接超时的报错，通常原因是迁移用的设备非ECS实例或不是与OSS同地域的ECS实例，但是配置文件使用了OSS的内网域名。OSS内网域名仅支持同地域ECS实例访问。
- 解决方法：

- 修改配置文件中域名为外网Endpoint，清除任务后重新提交任务。
- 使用与OSS同地域的ECS实例运行迁移任务。

迁移失败日志报错 `The specified bucket is not valid.`

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

- 问题原因：dest Domian配置错误。
- 解决方法：dest Domian配置的域名是Bucket所在地域的Endpoint地址，而不是带Bucket名称的二级域名。例如Bucket所在地域为华北2（北京），应填写oss-cn-beijing.aliyuncs.com。详情请参见[配置文件示例](#)。

迁移失败日志报错 `Unable to execute HTTP request: The Difference between ... is too large.`

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

- 问题原因
 - 本地机器时间不对，与OSS服务器时间相差15分钟以上，该情况居多。
 - 可能是并发太高，尤其是CPU占用率很高，导致并发上传慢。
- 解决方法
 - 修改本地机器时间，与OSS服务器一致。
 - 如果是并发问题，可以通过降低sys.properties文件的workerTaskThreadNum参数调整并发。

迁移失败日志报错 `No route to host.`

- 问题原因：可能本地防火墙或者iptables等原因导致网络不通。
- 解决方法：通过ping命令测试迁移服务器到源端和目的端网络是否正常。
 - 如果网络正常，检查电脑防火墙和本地的防火墙设备是否有限制，可尝试关闭防火墙进行测试。
 - 如果网络异常，排查网络原因进行相关处理后重试。

使用HTTP模式迁移，迁移失败日志报错 `Unknown http list file format.`

- 问题原因：指定的HTTP列表文件格式错误或内容不符合规范。
- 解决方法：
 - 如果是从其他操作系统上拷贝的文件，Linux系统可以用mac2unix、dos2unix等相关命令转换文件格式；Windows系统可以使用notepad等工具转换格式。
 - 如果是列表文件内容的格式不正确，请修改为正确的格式。关于列表文件内容的格式，请参见[说明及配置](#)。

迁移失败日志报错 `The object key "/xxxxx.jpg" is invalid.`

```
Exception:java.lang.IllegalArgumentException: The object key "/xxxxx.jpg" is invalid. An object name should be between 1 - 1023 bytes long when encoded as UTF-8 and cannot contain LF or CR or unsupported chars in XML1.0, and cannot begin with "/" or "\".
```

- 问题原因：srcPrefix或者destPrefix填写错误。
- 解决方法
 - 检查srcPrefix是否为目录。如果是目录，请以正斜线 (/) 结尾。
 - 检查destPrefix是否以正斜线 (/) 或者反斜线 (\) 开头。如果是，请删除正斜线 (/) 或反斜线 (\)。

任务运行常见问题

如果在迁移运行过程中出现问题，您可以先查看任务运行日志。

- 单机部署：运行日志的文件路径为 *logs/ossimport2.log*。
- 分布式部署：运行日志的文件路径为 *logs/import.log*。

执行任何命令均报错 `UnsupportedClassVersionError`。

```
Exception in thread "main" java.lang.UnsupportedClassVersionError: com/aliyun/ossimport2/OS
SImport2 : Unsupported major.minor version 51.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClassCond(ClassLoader.java:631)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:615)
    at com.simontuffs.onejar.JarClassLoader.defineClass(JarClassLoader.java:693)
    at com.simontuffs.onejar.JarClassLoader.findClass(JarClassLoader.java:599)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
    at com.simontuffs.onejar.Boot.run(Boot.java:300)
    at com.simontuffs.onejar.Boot.main(Boot.java:159)
```

- 问题原因：Java版本过低。
- 解决方法：将Java版本升级为1.7或1.8版本。

使用submit命令提交任务报错 `InvocationTargetException`。

```
Exception in thread "main" java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:497)
    at com.simontuffs.onejar.Boot.run(Boot.java:306)
    at com.simontuffs.onejar.Boot.main(Boot.java:159)
Caused by: java.lang.NullPointerException
    at com.aliyun.ossimport2.config.JobConfig.load(JobConfig.java:44)
    at com.aliyun.ossimport2.OSSImport2.doSubmitJob(OSSImport2.java:289)
    at com.aliyun.ossimport2.OSSImport2.main(OSSImport2.java:120)
    ... 6 more
```

- 问题原因：系统检查配置文件时发现配置文件不完整，例如某些配置项被删除或已注释。

- 解决方法：还原配置文件中被删除或注释的配置项。对于不需要配置的内容请在等号后留空删除配置项。配置示例请参见[配置文件示例](#)。

任务运行日志报错 `com.aliyun.oss.ClientException: Unknown.`

```
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.
```

- 问题原因：`com.aliyun.oss.ClientException: Unknown.` 以及 `SocketTimeoutException` 错误通常是网络被占满。
- 解决方法：ossimport会自动进行重试，如果重试完后仍然失败，您可以在任务完成后调用`retry`命令再次重试。

Linux系统运行日志报错 `too many open files.`

解决方法：使用`ulimit -n`查看系统句柄。

- 如果值在10,000以下，先通过`ulimit -n 65536`增大该值，然后重启进程。
- 如果值在10,000以上，先通过`sudo lsof -n`排查是哪些进程打开了句柄。然后评估这些进程是否需要保留。如果不需要请释放相应的句柄。

Windows系统启动任务后秒退

- 问题原因
 - 没安装Java或Java版本低于1.7。
 - 配置文件错误。
- 解决方法
 - 安装1.8版本的Java。
 - 按照示例正确编辑配置文件。配置示例，请参见[配置文件示例](#)。

使用submit命令提交任务后，通过stat命令查看任务状态一直显示 `no jobs is running or finished.`

```
bash console.sh stat
[WARN] List files dir not exist : /home/<user>/ossimport/workdir/master/jobs/
no jobs is running or finished.
```

- 问题原因：任务开始运行时，未启动服务。任务已提交且服务已启动后，才能通过`stat`命令才能查看到任务的运行状态。
- 解决方法
 - 使用`start`命令启动服务。
 - 如果服务已启动，且任务刚提交，Master需要先去扫描文件列表，这时Task还没有真正生成和分发，因此出现该报错是正常现象。
 - 如果服务已启动，且任务已提交一段时间后仍然出现该错误，需查看该进程启动后是否异常退出。如果是单机部署，请通过`logs/ossimport2.log`查看日志文件。如果是分布式部署，请通过`logs/ossimport.log`查看日志文件。找到异常原因并解决，然后再启动服务进程。

使用stat命令查看任务状态一直显示 `scanFinished: false.`

解决方法：观察Task的总数是否增加。

- 如果Task总数增加，则表明Job的文件列表还在List新的文件，是正常现象。
- 如果Task总数没变化，且Job配置的是增量模式，scanFinished不为true的情况下，ossimport会根据您配置的间隔时间定期扫描文件列表，检查有无新增或修改的文件。
- 如果不是增量模式，Task数不会增多，此时应检查运行日志是否异常。如果是单机部署，请通过`logs/ossimport2.log`查看日志文件。如果是分布式部署，请通过`logs/ossimport.log`查看日志文件。找到异常原因并解决，然后再启动服务进程。

Linux系统服务进程异常，日志却没有输出异常

- 问题原因：如果机器的可用内存少于，服务进程可能因为内存不足出现异常。
- 解决方法：请检查dmesg日志是否存在因内存不足导致进程异常的记录。

进程异常后，重启服务时需要执行哪些操作

解决方法：如果没有使用clean命令清除同名任务，所有提交过的Job都会有断点记录。您可以直接调用start命令启动服务即可，已经提交的Job不需要重新提交。

如何将Linux系统中文件名乱码的文件上传到OSS

解决方法：

1. 确认乱码文件的编码格式。
2. 使用`export LANG=<your file name encode>`命令解析编码。
3. 使用clean命令清除原来的Job，然后使用submit命令重新提交Job。

启动服务时报错 `java.nio.file.AccessDeniedException.`

- 问题原因：没有权限访问配置文件。
- 解决方法
 - 将配置文件的权限设置为所有人可读。
 - 将操作系统账号切换为管理员后启动服务。

Task个数显示为0，但JobState显示成功

```
[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
```

- 问题原因
 - srcPrefix填写错误，导致列举不出来文件。
 - srcPrefix下只有目录，没有文件，因为目录概念是OSS模拟出来的，不会被真正上传。
- 解决方法：填写正确的srcPrefix参数，并确保srcPrefix中有可用文件。

提交任务时报错 `InvocationTargetException.`

```
submit job:/disk2/ossimport2/local_job.cfg
Exception in thread "main" java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.simontuffs.onejar.Boot.run(Boot.java:306)
    at com.simontuffs.onejar.Boot.main(Boot.java:159)
Caused by: java.lang.NullPointerException
    at com.aliyun.ossimport2.OSSImport2.doSubmitJob(OSSImport2.java:289)
    at com.aliyun.ossimport2.OSSImport2.main(OSSImport2.java:120)
    ... 6 more
```

- 问题原因：配置文件错误或配置文件路径错误。
- 解决方法
 - 正确配置`conf/sys.properties`中的配置项`workingDir`。
 - 检查配置文件路径是否正确。

同步过程中显示源端的文件不存在

问题原因：Master执行迁移任务时，会先List文件列表，然后按照文件列表迁移数据。如果List操作完成后，源端某些文件被删除，则出现源端文件不存在的情况。此时，Master会跳过已删除的源端文件，并在错误列表里输出此类文件。

Job配置文件正确，但进行迁移任务的时候，运行状态和Job配置文件不符。

- 问题原因：如果之前提交过迁移任务，任务被中止后，对Job配置文件的修改不生效。
- 解决方法：使用`clean`命令清除之前的任务，待Job配置文件修改完成后重新提交任务。

文件正常传输，但报错 `NullPointerException`。

- 问题原因：ossimport 2.3.5版本新增文件统计功能，该功能需要加载cpt文件，而HTTP任务的cpt加载格式与其他源不同，导致程序不匹配出现异常。
- 解决方法
 - 将ossimport回退至2.3.4版本。
 - 忽略该报错。该问题只是打印报错，不影响迁移任务。

又拍云迁移常见问题

又拍云迁移的任务数一直显示0

解决方法：查看任务运行日志。

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

- 如果日志中 `recentLatency` 小于或等于30,000，表示正常List文件。又拍云List过程较慢，通常会超出30秒的超时时间。返回文件的个数取决于30秒内List的文件个数，请等待List操作完成。
- 如果 `recentLatency` 数值较低，通常是账号密码填写错误。又拍云的SDK出错只返回null，不返回错误结果。这种情况下，需要通过抓包的方式获取又拍云返回的错误码进行排查。

又拍云迁移时如何填写 `srcAccessKey` 和 `srcSecretKey`

解决方法：填写又拍云操作员的账号和密码。

又拍云迁移时一直显示HTTP 429错误

- 问题原因：又拍云对SDK访问间隔进行了限速处理。
- 解决方法：请联系又拍云解除限速限制。

任务完成OSS控制台显示的数据量比源数据量小

- 问题现象：Job全部成功上传完后，OSS控制台里显示Bucket大小无变化，但是Linux系统下使用`du`命令统计的大小与数据量实际大小相差较大。
- 问题原因
 - OSS控制台的Bucket数据量会延迟1~2小时更新，请在1~2小时后查看Bucket大小是否有变化。
 - Linux系统的`du`命令统计的是数据块大小，数据块大小大于数据量实际大小。建议使用 `ls -lR <目录绝对路径> | grep "\-rw" | awk '{sum+= $5}END{print sum}'` 命令统计本地目录的实际大小。

Linux系统执行时命令时出现 `unknown command "java"`、`unknown command "nohup"` 等提示

- 问题原因：没有安装对应命令。
- 解决方法：请使用`yum`、`apt-get`或`zypper`命令安装相应的命令。

配置文件中的 `srcPrefix` 是否支持指定为文件格式

不支持。`srcPrefix` 只支持目录或者前缀级别。

ossimport是否支持设置代理

不支持。

OSS之间的迁移为什么还会产生费用

如果配置了内网域名，将不收取流量费用，但仍然收取访问次数的费用。

开启增量模式后，OSS是否会同步删除本地已删除的文件

不删除。本地删除操作不会同步到OSS。

开启增量模式后，本地新增的文件没有被同步到OSS

增量模式通过对比文件最后修改时间来判断文件是否为增量。Linux的`mv`，Windows的`cp`、`mv`以及`rsync`带 `-t` 或者 `-a` 参数等操作不会修改文件的最后修改时间，因此通过此类操作修改的文件不会被扫描，也不会同步迁移至OSS。

迁移文件到OSS时是否支持同步迁移文件权限

不支持。您可以在迁移完成后使用命令行工具`ossutil`的`set-meta`命令修改文件权限。具体操作，请参见[set-meta（管理文件元信息）](#)。

5.ossfs

5.1. 概述

ossfs能让您在Linux系统中，将对象存储OSS的存储空间（Bucket）挂载到本地文件系统中，您能够像操作本地文件一样操作OSS的对象（Object），实现数据的共享。

运行环境

ossfs基于fuse用户态文件系统开发，只能运行在支持fuse的机器上。OSS提供了Ubuntu和CentOS系统的安装包，如果需要在其它环境下运行，可以通过源码方式构建目标程序。

ossfs支持在阿里云内网以及互联网环境下使用。在内网环境下时，建议使用内网访问域名，以提升访问速度和稳定性。

安装ossfs

- ossfs的下载及安装操作，请参见[快速安装](#)。
- ossfs的配置操作，请参见[高级配置](#)。

主要功能

ossfs基于s3fs构建，具有s3fs的全部功能。其中包括：

- 支持POSIX文件系统的大部分功能，例如上传下载文件、目录，设置用户权限等。
- 默认使用OSS的分片上传和断点续传功能上传文件。
- 支持MD5校验，保证数据完整性。

使用限制

- 不支持挂载归档型Bucket。
- 不适合高并发读写场景。
- 编辑OSS内文件会导致文件被重新上传。
- 元数据操作（例如list directory）需要远程访问OSS服务器，所以性能较差。
- 重命名文件或文件夹可能会出错。若操作失败，可能会导致OSS和本地数据不一致。
- 多个客户端挂载同一个OSS Bucket时，数据一致性由您自行维护。建议您合理规划文件使用时间，避免出现多个客户端写同一个文件的情况。
- 不支持hard link。
- 如果您因意外中断了文件上传的过程，且未继续完成该文件的上传，则已上传的部分会以碎片（Part）的形式存储在OSS的存储空间（Bucket）中。如果您不再需要这些Part，建议您通过以下方式删除，以免产生额外的存储费用。
 - 手动删除Part，请参见[管理碎片](#)。
 - 通过生命周期规则自动删除Part，请参见[设置生命周期规则](#)。

 **注意** 由于数据需要经过网络同步到云端，ossfs在性能和功能上可能与本地文件系统有差距。如果您想让数据库等对I/O要求很高的应用在ossfs挂载的磁盘上运行，需慎重考虑。

5.2. 快速安装

ossfs允许您在Linux系统中将对象存储OSS的存储空间（Bucket）挂载到本地文件系统。挂载完成后，您能够像操作本地文件一样操作OSS的对象（Object），从而实现数据共享。

运行环境

建议您在以下环境中运行ossfs：

- Linux系统
 - CentOS 7.0及以上版本
 - Ubuntu 14.04及以上版本
- fuse 2.8.4以上版本

下载地址

Linux发行版	下载
Ubuntu 18.04 (x64)	ossfs_1.80.6_ubuntu18.04_amd64.deb
Ubuntu 16.04 (x64)	ossfs_1.80.6_ubuntu16.04_amd64.deb
Ubuntu 14.04 (x64)	ossfs_1.80.6_ubuntu14.04_amd64.deb
CentOS 8.0 (x64)	ossfs_1.80.6_centos8.0_x86_64.rpm
CentOS 7.0 (x64)	ossfs_1.80.6_centos7.0_x86_64.rpm

以上列表中提供了常用系统的安装包，如果您的系统版本不在以上列表，请通过源码编译。源码地址请参见[GitHub ossfs](#)。

 **注意** 复制下载链接时会自动加上spm编码内容，使用wget命令下载时，请删除链接中的 `spm=xxxx`。

快速安装

1. 下载安装包。

以下载CentOS 7.0 (x64)版本为例：

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

2. 安装ossfs。

Ubuntu系统

以Ubuntu 16.04 (x64)版本为例，安装命令如下：

```
sudo apt-get update
sudo apt-get install gdebi-core
sudo gdebi ossfs_1.80.6_ubuntu16.04_amd64.deb
```

CentOS系统

以CentOS 7.0(x64)版本为例，安装命令如下：

```
sudo yum install ossfs_1.80.6_centos7.0_x86_64.rpm
```

对于使用yum安装rpm包的客户端，如果客户端节点网络环境特殊，无法直接使用yum下载依赖包。您可以在网络正常的、相同版本操作系统的节点上，使用yum下载依赖包并拷贝到网络特殊的节点。例如，ossfs需要依赖fuse 2.8.4以上版本，可使用如下命令，下载yum源中最新的fuse到本地：

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

 **注意** 如果需要下载其他依赖包，请将fuse换成对应包的名称。

3. 配置账号访问信息。

将Bucket名称以及具有该Bucket访问权限的AccessKey ID和AccessKey Secret信息存放在`/etc/passwd-ossfs`文件中。文件的权限建议设置为640。

```
echo BucketName:yourAccessKeyId:yourAccessKeySecret > /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
```

4. 将Bucket挂载到指定目录。

```
ossfs BucketName mountfolder -o url=Endpoint
```

将杭州地域名称为 `bucket-test` 的Bucket挂载到 `/tmp/ossfs` 目录下。

```
echo bucket-test:LTAIbZcdVCmQ****:Mok8x0y9hxQ3lcoh7A5e2MZEUz**** > /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
mkdir /tmp/ossfs
ossfs bucket-test /tmp/ossfs -o url=http://oss-cn-hangzhou.aliyuncs.com
```

 **注意** 如果您使用从阿里云购买的云服务器ECS来提供ossfs服务，您可以使用内网域名。例如，在该示例中您可以将OSS Endpoint 修改为 `oss-cn-hangzhou-internal.aliyuncs.com`，从而节省流量费用。有关OSS内网域名的更多信息，请参见[访问域名和数据中心](#)。

5. 如果您不希望继续挂载此Bucket，您可以将其卸载。

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

更多信息，请参见[GitHub ossfs](#)。

版本日志

有关版本日志的更多信息，请参见[GitHub ChangeLog](#)。

5.3. 高级配置

本文介绍ossfs的一些配置操作，您可以通过本文了解如何更好的使用ossfs。

前提条件

已安装ossfs，安装步骤请参见[快速安装](#)。

配置账号信息

通过ossfs访问OSS存储空间时，需要配置账号信息，也就是AccessKeyId和AccessKeySecret。这些账号信息需要按照特定的格式写到账号配置文件中。当挂载ossfs时，会从这个账号配置文件上获取账号信息，格式为 `$bucket_name:$access_key_id:$access_key_secret`。

账号配置文件的默认路径为 `/etc/passwd-ossfs`，您也可以通过 `-opasswd_file=passwd-path` 选项指定配置文件。两者的区别在于：默认路径的权限可以是640，其他路径下的配置文件权限必须是600。

- 同一个账号配置文件里可以保存多条账号信息，一条记录一行。ossfs会根据挂载的存储空间名称匹配到正确的账号上。

配置示例：

```
echo bucket-test-1:AAAIbZcdVCmQ****:AAA8x0y9hxQ3lcoh7A5e2MZEUz**** > /etc/passwd-ossfs
echo bucket-test-2:BBBIbZcdVCmQ****:BBB8x0y9hxQ3lcoh7A5e2MZEUz**** >> /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
mkdir /tmp/ossfs-1
mkdir /tmp/ossfs-2
ossfs bucket-test-1 /tmp/ossfs-1 -ourl=http://oss-cn-hangzhou.aliyuncs.com
ossfs bucket-test-2 /tmp/ossfs-2 -ourl=http://oss-cn-hangzhou.aliyuncs.com
```

- 当需要同时挂载多个存储空间时，您可以将所有的配置信息写到同一个账号配置文件里，也可以将不同的账号信息写到不同的账号配置文件中，通过 `-opasswd_file=xxx` 选项加载。

配置示例：

```
echo bucket-test-3:CCCIbZcdVCmQ****:CCC8x0y9hxQ3lcoh7A5e2MZEUz**** > /etc/passwd-ossfs-3
chmod 600 /etc/passwd-ossfs-3
mkdir /tmp/ossfs-3
ossfs bucket-test-3 /tmp/ossfs-3 -ourl=http://oss-cn-hangzhou.aliyuncs.com -opasswd_file=
/etc/passwd-ossfs-3
echo bucket-test-4:DDDIbZcdVCmQ****:DDD8x0y9hxQ3lcoh7A5e2MZEUz**** > /etc/passwd-ossfs-4
chmod 600 /etc/passwd-ossfs-4
mkdir /tmp/ossfs-4
ossfs bucket-test-4 /tmp/ossfs-4 -ourl=http://oss-cn-hangzhou.aliyuncs.com -opasswd_file=
/etc/passwd-ossfs-4
```

使用实例RAM角色访问

在云服务器ECS上，您还可以通过实例RAM角色的方式挂载ossfs。实例RAM角色允许您将一个角色关联到云服务器实例，在实例内部基于临时凭证STS访问OSS。临时凭证由系统自动生成和更新，应用程序可以使用指定的实例元数据URL获取临时凭证，无需特别管理。借助于RAM，一方面保证AccessKey安全，另一方面实现权限的精细化控制和管理。实例RAM角色详情请参见[实例RAM角色](#)。

下面以角色名为EcsRamRoleOssTest的RAM角色为例，来说明如何通过实例RAM角色挂载ossfs。

1. 创建名为EcsRamRoleOssTest的实例RAM角色。

配置步骤请参见[步骤一：创建实例RAM角色](#)。

2. 为实例RAM角色授予访问OSS访问权限。

配置步骤请参见[为RAM角色授权](#)。本示例授予实例RAM角色AliyunOSSReadOnlyAccess（只读访问OSS）权限，实际环境中，您可以自定义访问权限，详情请参见[创建自定义权限策略](#)。

3. 为实例授予RAM角色。

配置步骤请参见[步骤三：为实例授予RAM角色](#)。

4. 通过实例元数据URL挂载ossfs。

- i. 登录ECS实例。

- ii. 挂载ossfs，并增加-oram_role选项。

以位于杭州，名为Bucket1的Bucket为例：

```
ossfs bucket1 /tmp/ossfs -ourl=http://oss-cn-hangzhou.aliyuncs.com -oram_role=http://100.100.100.200/latest/meta-data/ram/security-credentials/EcsRamRoleOssTest
```

配置访问权限

ossfs挂载的目录访问权限默认为挂载点的所有者，即执行挂载命令的用户，其他用户无法访问。如果要修改默认的权限设置，例如允许其他用户或用户组访问挂载点，可以在运行ossfs的时候使用如下参数，做到期望的权限设置。

- allow_other: 赋予计算机上其他用户访问挂载目录的权限，但不包括目录内的文件。如果您要更改文件夹中的文件访问权限，请用chmod命令。该选项不需要设置选项值，如果需要启用，请直接添加-oallow_other选项。
- uid: 设置文件夹属于某个用户时填写的用户uid。
- gid: 设置文件夹属于某个用户时填写的用户gid。
- mp_umask: 用来设定挂载点的权限掩码，只有当allow_other选项设置后，该选项才生效，默认值为000。使用方法与umask命令使用方式一致。例如需要设置挂载点的权限为770，则增加-oallow_other-omp_umask=007；需要设置挂载点的权限为700，则增加-oallow_other-omp_umask=077。

配置示例：

- 允许所有用户访问，即权限为777。

```
ossfs bucket_name mount_point -ourl=endpoint -oallow_other
```

- 只允许同组用户访问，即权限为770。

```
ossfs bucket_name mount_point -ourl=endpoint -oallow_other -omp_umask=007
```

- 挂载时指定为其他用户和组，同时只允许同组的用户访问，即权限为770。

以www用户为例说明，先通过id命令获取用户的uid和gid信息，之后在挂载时指定uid和gid参数。

```
id www
uid=1000(www) gid=1000(web) groups=1000(web)
ossfs bucket_name mount_point -ourl=endpoint -oallow_other -ouid=1000 -ogid=1000 -omp_umask=007
```

挂载指定文件目录

ossfs除了可以把整个存储空间挂载到本地文件系统外，还可以通过设置前缀，把存储空间下的某个文件目录挂载到本地文件系统。命令格式如下：

```
ossfs bucket:/prefix mount_point -ourl=endpoint
```

通过这个方式挂载时，需要确保存储空间里存在 $\$(prefix)/$ 这样一个对象。可以通过ossutil的stat（[查看Bucket和Object信息](#)）命令查询该对象是否存在。

示例：将位于杭州地域的存储空间bucket-ossfs-test下的folder目录挂载到/tmp/ossfs-folder下。

```
ossfs bucket-ossfs-test:/folder /tmp/ossfs-folder -ourl=http://oss-cn-hangzhou.aliyuncs.com
```

开机自动挂载目录

1. 将Bucket名称、AccessKeyID、AccessKeySecret等信息写入`/etc/passwd-ossfs`文件，并将文件权限修改为640。

配置步骤请参见[快速安装](#)。

2. 设置开机自动挂载。

针对不同的系统版本，设置开机自动挂载ossfs的方式有所不同。

- o Ubuntu 14.04和CentOS 6.5系统版本通过fstab的方式自动挂载

- a. 在`/etc/fstab`中加入如下命令：

```
ossfs#bucket_name mount_point fuse _netdev,url=url,allow_other 0 0
```

- b. 保存`/etc/fstab`文件。执行`mount -a`命令，如果没有报错，则说明设置正常。

- c. 上一步执行完成后，Ubuntu 14.04就能自动挂载了。CentOS 6.5还需要执行如下命令：

```
chkconfig netfs on
```

- o CentOS 7.0及以上的系统通过开机自动启动脚本进行挂载

- a. 在`/etc/init.d/`目录下建立文件`ossfs`，将[模板文件](#)中的内容拷贝到这个新文件中。并将其中的`your_xxx`内容改成您自己的信息。

- b. 为新建立的`ossfs`脚本赋予可执行权限：

```
chmod a+x /etc/init.d/ossfs
```

命令执行完成后，您可以尝试执行该脚本，如果脚本文件内容无误，那么此时OSS中的Bucket已经挂载到您指定的目录下了。

- c. 把`ossfs`启动脚本作为其他服务，开机自动启动：

```
chkconfig ossfs on
```

- d. 执行上述步骤后，ossfs就可以开机自动挂载了。

使用Supervisor启动ossfs

Supervisor是用Python开发的一套通用的进程管理程序，能将一个普通的命令行进程变为后台daemon，并监控进程状态。异常退出时能自动重启。使用Supervisor启动ossfs的步骤如下：

1. 安装supervisor。

- o CentOS系统

```
yum install supervisor
```

- o Ubuntu系统

```
sudo apt-get install supervisor
```

2. 创建ossfs的启动脚本。

- i. 创建`start_ossfs.sh`文件。

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

ii. 写入启动脚本。

```
# 卸载
fusermount -u /mnt/ossfs
# 重新挂载，必须要增加-f参数运行ossfs，让ossfs在前台运行。
exec ossfs bucket_name mount_point -ourl=endpoint -f
```

3. 编辑`/etc/supervisor/supervisord.conf`文件，在最后加入如下内容：

```
[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. 运行Supervisor。

```
supervisord
```

5. 确认运行正常。

```
ps aux | grep supervisor # 应该能看到Supervisor进程。
ps aux | grep ossfs # 应该能看到ossfs进程。
kill -9 ossfs # 关闭ossfs进程，Supervisor应该会重启它。不要使用killall，因为killall发送SIGTERM，进程正常退出，Supervisor不再去重新运行ossfs。
ps aux | grep ossfs # 应该能看到ossfs进程。
```

开启调试日志

在使用ossfs的过程中，可能会遇到一些问题。这个时候需要开启调试日志，通过日志信息分析和定位问题。您可以通过如下方式开启调试日志：

- 在挂载目录时增加`-d -odbglevel=debug -ocurldb`选项，ossfs会把日志写入系统日志中。
 - CentOS系统
日志保存在`/var/log/messages`中。
 - Ubuntu系统
日志保存在`/var/log/syslog`中。
- 在挂载目录时使用`-d -odbglevel=debug -ocurldb -f`选项，ossfs会把日志输出到屏幕上。

5.4. 查看选项

您可以通过`-h`选项来查看ossfs支持的设置参数。

命令格式

```
./ossfs -h
```

您需要在ossfs程序所在目录使用该命令，默认目录为`/usr/local/bin/`，以实际安装环境为准。

常用选项

ossfs是基于fuse用户态文件系统来实现的，所以除了ossfs自己的选项外，也支持fuse的选项。挂载存储空间（Bucket）时，您可以根据具体的业务场景，设置不同的启动选项。选项格式如下：

```
-o option_name[=option_value] 或者 -ooption_name[=option_value]
```

例如在挂载时指定uid和gid参数：

```
ossfs bucket_name mount_point -ourl=endpoint -ouid=uid -ogid=gid
```

ossfs常用选项如下：

- url：用于指定存储空间的访问域名，格式为 `url=endpoint`。当不指定具体的请求协议时，默认选择 HTTP。

示例：

```
-ourl=oss-cn-hangzhou.aliyuncs.com
-ourl=http://oss-cn-hangzhou.aliyuncs.com
-ourl=https://oss-cn-hangzhou.aliyuncs.com
```

- passwd_file：用于指定保存存储空间访问密钥的文件，默认为 `/etc/passwd-ossfs`。这个文件的权限必须正确设置，当密钥文件为 `/etc/passwd-ossfs` 时，权限可设置为640；当密钥文件不为 `/etc/passwd-ossfs` 时，权限需要设置为600。密钥文件内容的格式为 `${bucket}:${access-key-id}:${access-key-secret}`。

示例：

```
echo bucket-test:LTAIbZcdVCmQ***:MOk8x0y9hxQ3lcoh7A5e2MZEUz*** > /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
echo bucket-test:LTAIbZcdVCmQ***:MOk8x0y9hxQ3lcoh7A5e2MZEUz*** > /passwd-path/passwd-ossfs
chmod 600 /passwd-path/passwd-ossfs
-opasswd_file=/passwd-path/passwd-ossfs
```

- max_stat_cache_size：用于指定文件元数据的缓存空间可缓存多少个文件的元数据。单位是个，默认值为1000。当目录下文件比较多时，可以调整这个参数，加快ls的速度。如果要禁止使用元数据缓存，可以设置为0。
- allow_other：赋予计算机上其他用户访问挂载目录的权限，但不包括目录内的文件。如果您要更改文件夹中的文件访问权限，请用chmod命令。该选项不需要设置选项值，如果需要启用，请直接添加-allow_other选项。
- dbglevel：用于设置日志信息的级别，支持日志信息级别为critical、error、warn、info和debug，默认值为critical，例如需要开启info级别的日志，请添加-odbglevel=info选项。日志信息会写到系统日志中，例如在centos系统会写到 `/var/log/messages` 中。
- -f：以前台方式而非守护进程方式运行ossfs，在前台模式下，日志会输出到终端屏幕。该参数一般用于调试问题时使用。
- -d：用于打开日志信息，同时该选项也会传递到fuse。在ossfs里，该选项等效于-odbglevel=info。

选项列表

无特别说明时，选项的格式均为 `-ooption_name=option_value` 或 `-o option_name=option_value`。

- ossfs选项列表

选项名称	描述
default_acl	指定往OSS写文件时设置的权限，默认设置为private。可选值为： ◦ <i>private</i>: 私有 ◦ <i>public-read</i>: 公共读 ◦ <i>public-read-write</i>: 公共读写 关于ACL的更多说明请参见Object ACL。
retries	指定请求失败后的重试次数，默认值为2。
storage_class	指定往OSS写文件时设置的存储类型，默认设置为Standard。可选参数为： ◦ <i>Standard</i>: 标准存储 ◦ <i>IA</i>: 低频访问存储 ◦ <i>Archive</i>: 归档存储 关于存储类型的更多详情请参见存储类型介绍。
public_bucket	指定用户通过匿名方式访问存储空间，该选项仅对ACL为public-read-write的存储空间有效。可选参数为： ◦ <i>0</i>: 不使用匿名方式访问，默认设置为该项。 ◦ <i>1</i>: 使用匿名方式访问。
passwd_file	用于指定保存存储空间访问密钥的文件，默认为 <i>/etc/passwd-ossfs</i> 。
connect_timeout	指定连接的超时时间，单位为秒，默认值为300秒。
readwrite_timeout	指定读或者写请求的超时时间，单位为秒，默认值为60秒。
max_stat_cache_size	指定文件元数据的缓存数量，单位为个，默认值为1000，约额外消耗4MB大小的空间。
stat_cache_expire	指定文件元数据缓存的失效时间，默认不失效。
no_check_certificate	对服务端的证书不做校验，仅在使用HTTPS协议时有效，默认开启证书校验。该选项无选项值，使用时请直接添加- <i>no_check_certificate</i> 选项。
multireq_max	列举文件时，访问文件元信息时的最大的并发数，默认值为20。
parallel_count	以分片模式上传大文件时，分片的并发数，默认值为5。
multipart_size	以分片模式上传数据时分片的大小，单位是MB，默认值为10。该参数会影响最大支持的文件大小。分片模式上传时，最多的分片数为10000，默认值下，最大的支持文件为100G。如果需要支持更大的文件，根据需求调整这个值。
url	指定存储空间的访问域名。
mp_umask	用来设定挂载点的权限掩码，只有当allow_other选项设置后，该选项才生效，默认值为000。使用方法与umask命令使用方式一致。例如需要设置挂载点的权限为770，则增加-oallow_other -omp_umask=007；需要设置挂载点的权限为700，则增加-oallow_other -omp_umask=077。
enable_content_md5	进行上传操作时，是否需要设置CONTENT_MD5，默认不设置。该选项无选项值，使用时直接增加-oenable_content_md5选项。

选项名称	描述
ram_role	使用ram_role方式访问OSS。当使用这个方式访问OSS时，会忽略掉密钥文件的AK（AccessKeyID）、SK（AccessKeySecret）参数。
dbglevel	设置日志信息的级别，支持日志信息级别为critical、error、warn、info和debug。默认值为critical。
curldbg	打开libcurl的日志信息，默认是不打开。该选项无选项值，如果需要输出libcurl日志请直接增加-ocurldbg选项。

• fuse选项列表

选项名称	描述
allow_other	修改挂载点的权限，允许所有的用户访问。该选择默认情况下只允许root用户设置。该选项无选项值，使用时请直接添加-oallow_other选项。
uid	设置文件夹属于某个用户时填写的用户uid。
gid	设置文件夹属于某个用户时填写的用户gid。

5.5. 常见问题

本文介绍在使用ossfs时遇到的一些问题案例及解决方案。

ossfs 的报错信息中均包含message。排查问题时，需要收集这些message，并根据message判断问题。例如socket建连失败、HTTP响应的状态码4xx、5xx等，使用前先开启debug-log。

- 403错误是因权限不足，导致访问被拒绝。
- 400错误是用户的操作方法有误。
- 5xx错误一般和网络抖动以及客户端业务有关系。

 **说明** 如果使用ossfs不满足业务需求时，可以考虑使用ossutil。

- ossfs是将远端的OSS挂载到本地磁盘，如果对文件读写性能敏感的业务，不建议使用ossfs。
- ossfs的操作不是原子性，存在本地操作成功，但OSS远端操作失败的风险。

案例：使用yum/apt-get安装ossfs时出现“conflicts with file from package fuse-devel”报错

问题分析：系统中存在老版本的fuse，与ossfs里的依赖版本冲突。

解决方案：请先使用相关的包管理器卸载fuse，再重新安装ossfs。

案例：安装ossfs时出现“fuse: warning: library too old, some operations may not work”报错

问题分析：出现错误的原因是ossfs编译时所使用的libfuse版本比运行时链接到的libfuse版本高，这往往是用户自行安装了libfuse导致的。Cent OS-5.x和Cent OS-6.x系统中，阿里云提供的ossfs安装包包含了libfuse-2.8.4，如果在运行的时候环境中存在libfuse-2.8.3，并且ossfs被链接到了旧版本的fuse上，就会出现上述错误。

您可以通过`ldd $(which ossfs) | grep fuse`命令确认ossfs运行时链接的fuse版本，如结果是`/lib64/libfuse.so.2`，那么通过`ls -l /lib64/libfuse*`命令可以看到fuse的版本。

解决方案：让ossfs链接到正确的版本。

1. 通过`rpm -ql ossfs | grep fuse`命令找到libfuse的目录。
2. 如果结果是`/usr/lib/libfuse.so.2`，则通过`LD_LIBRARY_PATH=/usr/lib ossfs ...`命令运行ossfs。

案例：安装依赖库fuse报错

```
Total size: 6.0 M
Installed size: 6.0 M
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test

Transaction Check Error:
  file /sbin/mount.fuse from install of ossfs-1.80.3-1.x86_64 conflicts with file
  from package fuse-2.8.3-5.el6.x86_64
  file /usr/bin/fusermount from install of ossfs-1.80.3-1.x86_64 conflicts with
  file from package fuse-2.8.3-5.el6.x86_64
  file /usr/bin/ulockmgr_server from install of ossfs-1.80.3-1.x86_64 conflicts
  with file from package fuse-2.8.3-5.el6.x86_64

Error Summary
-----

[root@localhost 123]# rpm -qa |grep fuse
fuse-2.8.3-5.el6.x86_64
fuse-ntfs-3g-2013.1.13-2.el6.rf.x86_64
You have new mail in /var/spool/mail/root
[root@localhost 123]#
```

问题分析：fuse的版本不满足ossfs的要求。

解决方案：手动下载fuse最新版本安装，不要使用yum安装。详情请参见[fuse](#)。

案例：挂载时出现“ossfs: unable to access MOUNTPOINT /tmp/ossfs: Transport endpoint is not connected”报错

问题分析：未创建该目录导致的。

解决方案：先创建对应的目录，之后再进行挂载操作。

案例：挂载时出现“fusermount: failed to open current directory: Permission denied”报错

问题分析：fuse的Bug，要求当前用户对当前目录（非挂载目录）有读权限。

解决方案：通过cd命令切换到到一个有读权限的目录，再运行ossfs命令。

案例：挂载时出现“ossfs: Mountpoint directory /tmp/ossfs is not empty. if you are sure this is safe, can use the 'nonempty' mount option”报错

问题分析：默认情况下，ossfs只能挂载到空目录下。当试图挂载到非空目录下时，会提示上述错误。

解决方案：切换到一个空目录下重新挂载。如果还是需要挂载到该目录下，挂载时增加`-ononempty`参数。

案例：挂载成功后，ls目录时出现“operation not permitted”报错

问题分析：请检查您的Bucket中，是否存在名称含有不可见字符的Object。文件系统对文件名和目录名有严格的限制，因此会收到上述错误。

解决方案：用其他工具对这些Object重命名后，ls就能正确显示目录内容了。

案例：ossfs 偶尔出现断开的情况

```
Nov 8 02:58:28 i2wz9exgcy43hdoulltcoez kernel: [30273] 0 30273 33017 827 61 0 0 AliYunDun
Nov 8 02:58:28 i2wz9exgcy43hdoulltcoez kernel: [16398] 0 16398 1301998 317459 955 0 0 java
Nov 8 02:58:28 i2wz9exgcy43hdoulltcoez kernel: Out of memory: Kill process 3017 (ssfs) score 58) or sacrifice child
Nov 8 02:58:28 i2wz9exgcy43hdoulltcoez kernel: Killed process 3017 (ssfs) total-vm:6840372kB, anon-rss:4787892kB, file-rss:208kB, shm-rss:0kB
Nov 8 02:58:45 i2wz9exgcy43hdoulltcoez s3fs[2366]: s3fs.cpp:lst_bucket(2459): xReadMemory returns with error.
Nov 8 02:58:45 i2wz9exgcy43hdoulltcoez s3fs[2968]: s3fs.cpp:s3fs_readdir(2382): lst_bucket returns error(-1).
Nov 8 02:58:45 i2wz9exgcy43hdoulltcoez s3fs[2968]: s3fs.cpp:lst_bucket(2459): xReadMemory returns with error.
Nov 8 02:58:45 i2wz9exgcy43hdoulltcoez s3fs[2968]: s3fs.cpp:s3fs_readdir(2382): lst_bucket returns error(-1).
Nov 8 02:59:01 i2wz9exgcy43hdoulltcoez systemd: Started Session 46140 of user root.
```

问题分析:

1. 开启ossfs的debug日志，加上-d -o f2参数，ossfs会把日志写入到系统 `/var/log/message`。
2. 分析日志，发现ossfs在list bucket、list object申请内存过多，触发了系统的oom。

④ 说明 list object是发起HTTP请求到OSS获取文件的meta信息，如果客户的文件很多，ls会消耗系统大量内存来获取文件的meta。

解决方案:

- 通过`-omax_stat_cache_size=xxx`参数增大stat cache 的 size，这样第一次ls会较慢，但是后续的ls速度会提高，因为文件的元数据都在本地cache中。这个值默认是1000，约消耗4 MB内存，请根据您的机器内存大小调整为合适的值。
- `ossfs`在读写时会占用磁盘写大量的temp cache，和Nginx差不多，可能会导致磁盘可用空间不足，需要经常清理。
- 使用`ossutil`替代`ossfs`，非线上敏感业务可以使用`ossfs`，要求可靠性、稳定性的建议使用`ossutil`。

案例：访问出现 “The bucket you are attempting to access must be addressed using the specified endpoint” 报错

问题分析：根据Message判断，错误原因是Endpoint指定错误，有以下两种可能性：

- Bucket和Endpoint不匹配。
- Bucket所属UID和实际的AccessKey对应的UID不一致。

解决方案：确认正确的配置信息并修改。

案例：通过cp命令拷贝数据时出现“input/output error”报错

[illegible]

问题分析: input/output error都是捕获到系统磁盘的错误而产生的报错, 可以查看出现报错时, 磁盘读写是否存在高负载的情况。

解决方案：可以增加分片参数，控制文件读写。使用`ossfs -h`命令可以查看分片参数。

案例：使用rsync同步时出现“input/output error”报错

```
cp: writing `/data/tmp/I_201704/12/request.1491926430217.avro': Input/output error
cp: closing `/data/tmp/I_201704/12/request.1491926430217.avro': Input/output error
```

问题分析: ossfs与rsync同步使用本身会出现问题。此案例中, 用户对一个141 GB的大文件进行cp操作, 使磁盘读写处于非常高的负载状态, 从而产生此报错。

解决方案：如果想要将OSS文件下载到本地ECS，或者本地上传到ECS，可以通过ossutil的分片上传、下载进行操作。

案例：上传大文件时出现“there is no enough disk space for used as cache(or temporary)”报错

```
[root@653a8832c9eb4d8d0c02-02-750000-node5 ~]# ossfs --add-dk /mnt/ossfs/daily -o multipart_size=31457280 -ourl=http://oss-ap-southeast-3-internal.aliyuncs.com
ossfs: There is no enough disk space for used as cache(or temporary) directory by s3fs.
[root@653a8832c9eb4d8d0c02-02-750000-node5 ~]#
```

● 问题原因

磁盘空间小于 `multipart_size * parallel_count`。

`multipart_size`表示分片大小（默认单位为MB），`parallel_count`表示并发上传分片数量（默认值为5）。

● 问题分析

ossfs默认通过分片上传的方式上传大文件。上传时，ossfs会将临时缓存文件写入`/tmp`目录下，写入前需要先判断`/tmp`目录所在的磁盘可用空间是否小于 `multipart_size * parallel_count`。如果磁盘可用空间大于 `multipart_size * parallel_count`，则正常写入文件。如果磁盘可用空间小于 `multipart_size * parallel_count`，则出现本地磁盘可用空间不足的报错。

例如，磁盘可用空间为300 GB，待上传的文件为200 GB，但`multipart_size`设置为100000（100 GB），并发上传分片数量保持默认值5。此时，ossfs判断上传的文件大小为100 GB*5=500 GB，超出本地磁盘可用空间。

● 解决方法

在并发上传分片数量保持默认值5的情况下，设置合理的`multipart_size`：

- 如果磁盘可用空间为300 GB，待上传的文件为200 GB，则`multipart_size`设置为20。
- 如果磁盘可用空间为300 GB，待上传的文件为500 GB，则`multipart_size`设置为50。

案例：挂载成功后，touch文件时出现403报错

```
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] curl.cpp.PutHeadRequest(2278): [tpath=/fuse_hidden00000027000000006]
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] curl.cpp.prepare_url(4038): URL is http://oss-cn-hangzhou-internal.aliyuncs.com/pgback/fuse_hidden00000027000000006
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] curl.cpp.prepare_url(4062): URL changed is http://pgback-oss-cn-hangzhou-internal.aliyuncs.com/fuse_hidden00000027000000006
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] curl.cpp.PutHeadRequest(2369): [path=/touch/fuse_hidden00000027000000006]
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] curl.cpp.RequestPerform(1829): HTTP response code 403 was returned, returning EPERM
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21840][INF] s3fs.cpp.s3fs_flush(2127): [path=/tmp/rockwell/touch]
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21840][INF] fdcache.cpp.RowFlush(1302): [tpath=/path=/tmp/rockwell/touch][fd=6]
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] s3fs.cpp.s3fs_release(2165): [path=/tmp/rockwell/touch][fd=6]
```

问题分析：403错误通常是访问权限问题导致的。以下情况会导致touch一个文件出现403报错。

- 该文件为归档类型文件，touch时会出现403报错。
- 使用的AccessKey无该存储空间的操作权限。

解决方案：

- 归档类型文件问题：将文件解冻后访问。
- 权限问题：为使用的AccessKey对应的账号配置正确的权限。

案例：使用ossfs上传到OSS的文件的Content-Type全是application/octet-stream

问题分析：上传文件时，ossfs通过查询`/etc/mime.types`中的内容来设置文件的Content-Type。当该文件不存在时，默认设置为`application/octet-stream`。

解决方案：请检查这个文件是否存在，如果不存在，则需要添加。

- 通过命令自动添加`mime.types`文件

- Ubuntu系统
使用`sudo apt-get install mime-support`命令添加。
- CentOS系统
使用`sudo yum install mailcap`命令添加。

- 手动添加 *mime.types* 文件

- i. 创建 *mime.types* 文件

```
vi /etc/mime.types
```

- ii. 添加需要的格式，每种格式一行，每行格式为 `application/javascript js`。

添加完成后，需要重新挂载OSS。

目录下有非常多的文件时，为什么ls该目录很慢

问题分析：假设一个目录下有N个文件，那么ls该目录至少需要N次OSS HTTP requests。在文件非常多的时候，这可能造成严重的性能问题。

解决方案：通过 `-omax_stat_cache_size=xxx` 参数增大stat cache的size，这样第一次ls会较慢，但是后续的ls就快了，因为文件的元数据都在本地cache中。这个值默认是1000，大约消耗4 MB内存，请根据您的机器内存大小调整为合适的值。

为什么用ossfs看到的文件信息（例如大小）与其他工具看到的不一致

问题分析：ossfs默认会缓存文件的元信息（包括大小/权限等），这样就不需要每次ls的时候向OSS发送请求，加快速度。如果用户通过其他程序（例如SDK/官网控制台/ossutil等）对文件进行了修改，由于缓存的关系，ossfs没有及时更新，导致与其他工具看到的文件信息不一致。

解决方案：可以在挂载的时候加上参数`-omax_stat_cache_size=0`，禁用元数据缓存功能。每次ls时，都会向OSS 发送请求，获取最新的文件信息。

在ECS上挂载OSS，如何避免因后台程序扫描文件而产生费用

问题分析：程序扫描ossfs挂载的目录，会转换成向OSS的请求。如果请求次数很多，会产生费用。

解决方案：可以通过auditd工具查看是哪些进程扫描了OSS挂载的目录。具体步骤如下：

1. 安装auditd并启动。

```
sudo apt-get install auditd
sudo service auditd start
```

2. 将OSS挂载的目录设置为监视目录，例如挂载目录为 `/mnt/ossfs`。

```
auditctl -w /mnt/ossfs
```

3. 在auditlog中查看是哪些进程访问了这个目录。

```
ausearch -i | grep /mnt/ossfs
```

4. 修改参数，跳过程序扫描。

例如通过auditlog查到是 `updatedb` 扫描了所挂载的目录，可以通过修改 `/etc/updatedb.conf` 让它跳过。具体做法是：

- i. 在 `RUNEFS =` 后面加上 `fuse.ossfs`。

- ii. 在 `PRUNEPATHS =` 后面加上挂载的目录。

6.ossftp

6.1. 概述

ossftp是一个特殊的FTP server，可以将对文件、文件夹的操作映射为对OSS的操作，使您可以基于FTP协议来管理存储在OSS上的文件。

使用限制

- ossftp主要提供给个人测试时使用，生产环境建议使用[OSS管理控制台](#)、[ossutil](#)、[ossbrowser](#)、[SDK](#)等工具管理您的OSS。
- FTP协议是明文传输的，为了防止您的密码泄漏，建议将ossftp和客户端运行在同一台机器上，通过 `127.0.0.1:port` 访问。
- ossftp在同一时间只允许一台客户端连接，新发起的连接请求会导致已经连接的客户端断开。

部署环境

- 支持系统：Windows、Linux、macOS
- 支持架构：x86（32bit、64bit）
- 运行环境：Python2.7、Python3.x

主要功能

- 支持文件和文件夹的上传、下载、删除操作。
- 支持使用分片上传方式上传大文件。
- 支持大部分FTP指令。

下载地址

操作系统	说明	安装包
Windows	Windows系统默认未安装Python2.7，所以安装包中包含了Python2.7，解压后即可使用。	ossftp-1.2.0-win.zip
Linux	Linux系统默认安装Python2.7或Python3.x，所以安装包中不再包含可执行的Python，只包含了相关依赖库。	ossftp-1.2.0-linux-mac.zip
macOS	macOS系统默认安装Python2.7或Python3.x，所以安装包中不再包含可执行的Python，只包含了相关依赖库。	ossftp-1.2.0-linux-mac.zip

相关文档

- [安装ossftp](#)
- [创建自定义登录用户](#)

- [FileZilla如何上传本地站点文件到OSS](#)
- [Discuz如何存储远程附件到OSS](#)
- [Phpwind如何存储远程附件到OSS](#)
- [WordPress如何存储远程附件到OSS](#)
- [常见问题](#)

6.2. 安装ossftp

本文介绍如何安装ossftp。

操作步骤

1. 下载ossftp安装包。

操作系统	说明	安装包
Windows	Windows系统默认未安装Python2.7，所以安装包中包含了Python2.7，解压后即可使用。	ossftp-1.2.0-win.zip
Linux	Linux系统默认安装Python2.7或Python3.x，所以安装包中不再包含可执行的Python，只包含了相关依赖库。	ossftp-1.2.0-linux-mac.zip
macOS	macOS系统默认安装Python2.7或Python3.x，所以安装包中不再包含可执行的Python，只包含了相关依赖库。	ossftp-1.2.0-linux-mac.zip

2. 解压ossftp安装包。

 **注意** 安装包解压后的路径不能包含中文。

3. 运行ossftp。

操作系统	操作方式
Windows	双击 <code>start.vbs</code> 。  说明 如果双击没有反应，请升级IE浏览器的版本或将默认浏览器设置为其他浏览器。
Linux	执行以下命令。 <pre>bash start.sh</pre>
macOS	双击 <code>start.command</code> 。

ossftp运行后会默认打开本机的以下端口：

- TCP 2048端口：FTP服务端点，用于接收FTP请求。
- TCP 8192端口：Web服务端点，用于打开ossftp的图形化管理界面。

 **说明** 如果您需要将ossftp服务提供他人使用，在防火墙配置中开放TCP 2048端口和TCP 8192端口。

4. 配置ossftp。

i. 通过浏览器访问ossftp的图形化管理界面。

设备	访问域名
本地机器	http://127.0.0.1:8192
其他机器	http://Linux服务器IP:8192

ii. 在ossftp的图形化管理界面完成以下配置。

系统配置选项

开机自动启动

OFF

启动时弹出状态页

ON

显示托盘图标

ON

ossftp 监听地址 (重启生效)

127.0.0.1

ossftp 监听端口 (重启生效)

2048

ossftp 被动端口范围起始端口 (重启生效)

51000

ossftp 被动端口范围终止端口 (重启生效)

53000

ossftp 日志等级 (DEBUG, INFO, WARNING, ERROR, CRITICAL。重启生效)

INFO

Bucket endpoints (大部分情况无需设置, 如何设置请参考wiki。重启生效)

示例: bucket-a.oss-cn-hangzhou.aliyuncs.com, bucket-b.oss-cn-qingdao.aliyuncs.com

Language(cn/en)

cn

保存配置

重启

退出

参数	示例值	说明
ossftp监听地址	127.0.0.1	使用FTP服务的客户端IP。如果是在本机上运行客户端，保持默认。

参数	示例值	说明
ossftp监听端口	2048	ossftp的接收访问请求的端口。不与其他端口冲突时，保存默认。
ossftp被动端口范围起始端口	51000	ossftp回应访问请求的起始端口。不与其他端口冲突时，保持默认。
ossftp被动端口范围终止端口	53000	ossftp回应访问请求的结束端口。不与其他端口冲突时，保持默认。
ossftp日志等级	INFO	设置ossftp的日志输出等级。取值如下： ■ <i>DEBUG</i>: 记录细粒度信息事件，一般用于调试程序。 ■ <i>INFO</i>: 记录软件正常运行发生的事件。 ■ <i>WARNING</i>: 记录不会对系统造成影响的非正常事件。 ■ <i>ERROR</i>: 记录会对系统造成影响，但不影响系统稳定性的非正常事件。 ■ <i>CRITICAL</i>: 记录导致系统无法正常工作的非正常事件。
Bucket endpoints	examplebucket.oss-cn-hangzhou.aliyuncs.com	Bucket的访问域名。格式为 <code>BucketName.Endpoint</code> 。多个域名以英文逗号(,) 隔开。
Language	cn	ossftp的显示语言。

iii. 单击保存配置，然后单击重启使配置生效。



注意 不要单击退出，否则会导致ossftp停止运行。

6.3. 创建自定义登录用户

ossftp客户端可以使用拥有Bucket访问权限的AccessKey访问ossftp服务端，也可以使用拥有Bucket访问权限的自定义登录用户访问ossftp服务端。本文介绍如何在ossftp服务端为ossftp客户端创建自定义登录用户。

前提条件

已安装ossftp。具体操作，请参见[安装ossftp](#)。

操作步骤

1. 登录安装了ossftp的服务器。
2. 切换到ossftp的安装目录。
3. 修改`config.json`中的accounts。

配置示例如下：

```
{
  "modules":{
    "accounts":[
      {
        //填写拥有Bucket访问权限的AccessKey ID和AccessKey Secret。
        "access_id":"LTAI4FrJPUSoKm4JH*****",
        "access_secret":"Y6IoUOZReouXvWaXuwjvDch9*****",
        //填写目标Bucket名称。
        "bucket_name":"examplebucket",
        //填写Bucket内文件的访问路径。填写后，该账号仅可以访问指定路径的文件；置空则表示允许访问Bucket内所有文件。
        "home_dir":"examplefolder/",
        //自定义登录密码。
        "login_password":"password1",
        //自定义登录用户名。
        "login_username":"user1"
      },
      {
        //填写拥有Bucket访问权限的AccessKey ID和AccessKey Secret。
        "access_id":"LTAI4FrJPUSoKm4JH*****",
        "access_secret":"Y6IoUOZReouXvWaXuwjvDch9*****",
        //填写目标Bucket名称。
        "bucket_name":"examplebucket",
        //填写Bucket内文件的访问路径。填写后，该账号仅可以访问指定路径的文件；置空则表示允许访问Bucket内所有文件。
        "home_dir":"",
        //自定义登录密码。
        "login_password":"password2",
        //自定义登录用户名。
        "login_username":"user2"
      }
    ],
    "launcher":{
      "auto_start":0,
      "control_port":8192,
      "language":"cn",
      "popup_webui":1,
      "show_systray":1
    },
    "ossftp":{
      "address":"127.0.0.1",
      "bucket_endpoints":"",
      "log_level":"INFO",
      "passive_ports_start":51000,
      "passive_ports_end":53000,
      "port":2048
    }
  }
}
```

4. 重启ossftp服务使自定义登录用户生效。

i. 通过浏览器访问ossftp的图形化管理界面。

设备	访问域名
本地机器	http://127.0.0.1:8192
其他机器	http://Linux服务器IP:8192

ii. 在ossftp的图形化管理界面，单击重启。

 **注意** 不要单击退出，否则会导致ossftp停止运行。

6.4. 使用案例

6.4.1. FileZilla如何上传本地站点文件到OSS

FileZilla客户端是一款简单易用的FTP客户端工具。本文介绍如何使用FileZilla客户端将本地站点的文件上传到OSS。

前提条件

- 已创建公共读权限的存储空间。具体操作，请参见[创建存储空间](#)。
- 已安装ossftp。具体操作，请参见[安装ossftp](#)。
- 已下载FileZilla客户端。下载地址，请参见[FileZilla](#)。

操作步骤

1. 打开FileZilla客户端。
2. 在FileZilla控制台，完成以下配置。

参数	示例值	说明
主机	127.0.0.1	服务器IP地址。如果ossftp和FileZilla客户端在一台设备上，使用默认地址127.0.0.1。
用户名	Y6loUOZReouXvWaXuwjvDch9** ****/examplebucket	连接ossftp的用户名。用户名由拥有Bucket访问权限的AccessKey ID和Bucket名称组成，格式为 <code>AccessKey ID/Bucket名称</code> 。如何获取AccessKey ID，请参见 获取AccessKey 。  说明 您可以自定义登录用户名。操作步骤，请参见 创建自定义登录用户 。

参数	示例值	说明
密码	sRaQg0i3SH9EqYXJSL1hlaR*****	连接ossftp的密码。密码为拥有Bucket访问权限的AccessKey Secret。如何获取AccessKey Secret，请参见 获取AccessKey 。  说明 您可以自定义登录密码。操作步骤，请参见创建自定义登录用户。
端口	2048	ossftp配置的监听端口。默认为2048。

- 3. 单击**快速连接**。
- 4. 将本地站点的文件拖动到远程站点验证配置是否成功。

6.4.2. Discuz如何存储远程附件到OSS

网站远程附件功能是指通过FTP方式将用户上传的附件直接存储到远程的FTP服务器，目前Discuz论坛、PHPWind论坛、WordPress个人网站等都支持远程附件功能。本文介绍如何基于Discuz论坛存储远程附件。

前提条件

- 已创建公共读权限的存储空间。具体操作，请参见[创建存储空间](#)。
本文以华东1（杭州）名为test-hz-jh-002的Bucket为例。
- 已安装ossftp。具体操作，请参见[安装ossftp](#)。
- 已搭建Discuz论坛。下载地址，请参见[Discuz! 官方网站](#)。
本文以Discuz! X3.1版本为例。

操作步骤

- 1. 使用管理员账号登录Discuz站点。
- 2. 在管理界面单击**全局**，然后单击**上传设置**。
- 3. 单击**远程附件**，然后设置远程附件参数。

参数	示例值	说明
启用远程附件	是	是否启用远程附件。
启用SSL连接	否	是否启用SSL连接。
FTP服务器地址	127.0.0.1	运行ossftp的地址。如果是在本机上运行ossftp，填写127.0.0.1。
FTP服务器端口	2048	ossftp的接收访问请求的端口。默认为2048。

参数	示例值	说明
FTP帐号	Y6IoUOZReouXvWwXuwjvDch9** ****/examplebucket	连接ossftp的用户名。用户名由拥有Bucket访问权限的AccessKey ID和Bucket名称组成，格式为 <code>AccessKey ID/BucketName</code> 。如何获取AccessKey ID，请参见 获取AccessKey 。 ❓ 说明 您可以自定义登录用户名。操作步骤，请参见创建自定义登录用户。
FTP密码	sRaQg0i3SH9EqYXJSL1hlaR*****	连接ossftp的密码。密码为拥有Bucket访问权限的AccessKey Secret。如何获取AccessKey Secret，请参见 获取AccessKey 。 ❓ 说明 您可以自定义登录密码。操作步骤，请参见创建自定义登录用户。
被动模式（PASV）连接	是	选择是。
远程附件目录	.	FTP服务在OSS的指定路径创建远程附件的上传目录。半角句号（.）表示在Bucket的根目录下创建上传目录。
远程访问URL	https://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com	Bucket的访问域名。格式为 <code>BucketName.Endpoint</code> 。
FTP传输超时时间	0	设置为0表示使用服务器默认超时时间。

4. 单击**测试远程附件**确认配置是否正常。

5. 发帖验证配置是否成功。

- i. 发帖时上传图片附件。
- ii. 在图片上右键单击，然后选择**在新标签页中打开链接**。

如果图片的链接格式为 `http(s)://BucketName.Endpoint/path/filename`，则表示附件已正常上传。

例如本示例中，图片URL为 `https://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/forum/201512/18/171012mzvkkku2z3na2w2wa.jpg`

6.4.3. Phpwind如何存储远程附件到OSS

本文介绍如何基于Phpwind论坛存储远程附件。

前提条件

- 已开通OSS服务，并创建了一个公共读权限的存储空间（Bucket）。
 - 开通OSS服务请参见[开通OSS服务](#)。
 - 创建Bucket的步骤请参见[创建存储空间](#)。
- 已搭建phpwind论坛。

背景信息

网站远程附件功能是指将用户上传的附件直接存储到远端的存储服务器，一般是通过FTP的方式存储到远程的FTP服务器。目前Discuz论坛、phpwind论坛、Wordpress个人网站等都支持远程附件功能。

本文档测试所用phpwind版本为phpwind8.7。

配置步骤

1. 使用管理员账号登录phpwind站点。
2. 在管理界面单击全局 > 附件设置 > 附件设置。
3. 单击FTP设置，设置FTP选项。

配置项	说明
使用FTP上传	选择开启。
站点附件地址	填写Bucket的外网访问域名，格式为http://BucketName.Endpoint。测试所用Bucket名为test-hz-jh-002，属于杭州地域。所以这里填写的是http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com。关于访问域名的详情请参见 OSS访问域名使用规则 。
FTP服务器地址	即运行ossftp工具的地址，通常填写127.0.0.1即可。
FTP服务器端口	默认为2048。
FTP上传目录	填半角句号（.）即可，表示在Bucket的根目录开始创建附件目录。
FTP帐号	格式为AccessKeyID/BucketName。注意这里的正斜线（/）不是或的意思。
FTP密码	即AccessKeySecret。
超时时间[秒]	设置为10，如果10秒内没有返回请求结果，表示系统将会超时返回。

4. 发帖验证配置是否成功。
 - i. 发帖时上传图片附件。
 - ii. 在图片上右键单击，之后单击在新标签页中打开链接。

通过图中的URL，我们可以判断图片已经上传到OSS的test-hz-jh-002 Bucket。

6.4.4. WordPress如何存储远程附件到OSS

本文介绍如何基于WordPress个人网站存储远程附件。

前提条件

- 已开通OSS服务，并创建了一个公共读权限的存储空间（Bucket）。
 - 开通OSS服务请参见[开通OSS服务](#)。
 - 创建Bucket的步骤请参见[创建存储空间](#)。
- 已搭建WordPress个人网站。

背景信息

网站远程附件功能是指将用户上传的附件直接存储到远端的存储服务器，一般是通过FTP的方式存储到远程的FTP服务器。目前Discuz论坛、phpwind论坛、WordPress个人网站等都支持远程附件功能。

WordPress本身不支持远程附件功能，但是可以通过第三方的插件来做远程附件。本文档示例中所用WordPress版本为4.3.1，所用插件为Hacklog Remote Attachment。

配置步骤

1. 使用管理员账号登录WordPress站点。
2. 单击**插件**，之后在关键词栏输入**FTP**并按回车键。
3. 找到Hacklog Remote Attachment，单击**现在安装**。
4. 插件安装完成后单击**设置 > Hacklog远程附件**。
5. 在弹出的Hacklog远程附件选项对话框设置FTP服务信息。

配置项	说明
Ftp服务器	即运行ossftp工具的地址，通常填写127.0.0.1即可。
Ftp服务器端口	默认为2048。
Ftp用户名	格式为AccessKeyID/BucketName。注意这里的正斜线（/）不是或的意思。
Ftp密码	即AccessKeySecret。
FTP超时	默认设置为30秒即可。
远程基本URL	填写Bucket的外网访问域名，格式为http://BucketName.Endpoint。测试所用Bucket名为test-hz-jh-002，属于杭州地域。所以这里填写的是http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/wp。关于访问域名的详情请参见 OSS访问域名使用规则 。
FTP远程路径	设置附件在Bucket的存储路径。示例中填写wp表示所有附件都会存储在Bucket的wp目录下。远程基本URL须与FTP远程路径对应。
HTTP远程路径	填半角句号（.）即可。

6. 单击**保存**。
单击**保存**的同时会测试配置，测试结果会在页面上方显示。
7. 发布新文章验证配置是否成功。
 - i. 撰写新文章时单击**添加媒体**来上传附件。
 - ii. 单击**发布**，即可看到刚撰写的文章。

iii. 在图片上右键单击，选择在新标签页中打开链接。

通过图中的URL，我们可以判断图片已经上传到OSS的test-hz-jh-002 Bucket。

6.5. 常见问题

本文主要介绍在使用ossftp时可能遇到的问题及解决方案。

连接FTP Server时提示无法连接到服务器

- 问题现象
- 问题原因及解决方法
 - 问题原因：输入的AccessKey ID和AccessKey Secret有误。
解决方法：输入正确的AccessKey ID和AccessKey Secret信息后重试连接服务器。
 - 问题原因：所用的AccessKey信息为RAM用户的AccessKey，但RAM用户没有访问OSS的权限。
解决方法：请结合实际使用场景为RAM用户配置相应的权限，以下为常见场景的权限说明。

- 只读访问OSS某个Bucket的资源

所需权限为 `oss:ListObjects` 和 `oss:GetObject`

- 在OSS某个Bucket中写入数据

所需权限为 `oss:ListObjects` 和 `oss:PutObject`

- 删除Bucket中的数据

所需权限为 `oss:ListObjects` 和 `oss:DeleteObjects`

关于授权RAM用户其他场景的配置示例，请参见[RAM Policy常见示例](#)。

使用FileZilla连接FTP Server时报错501

- 问题现象
在Linux下运行FTP Server，然后使用FileZilla连接时报错501

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

- 问题原因
本地中文编码问题。
- 解决方法

i. 在将要运行start.sh的终端中输入下面的命令。

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

ii. 重新启动FileZilla。

登录ossftp成功后List文件超时导致连接断开

Bucket根目录下的文件或文件夹数量过多。登录ossftp后，FTP Server会尝试List Bucket根目录下的所有文件和文件夹。单次可以List 1000个文件和文件夹。如果根目录下文件和文件夹数量超过100万，会导致1000次以上的List请求，从而造成超时。

数据传输不成功

- 问题原因

FTP协议的控制端口与数据端口不同。当FTP Server在被动模式下需要传输数据时，FTP Server会打开1个随机端口用于连接客户端。当FTP Server所在机器有端口限制时，可能会导致数据无法正常传输。

- 解决方法

运行 `ftpserver.py` 时，通过指定 `--passive_ports_start` 和 `--passive_ports_end` 选项设置本地端口的起止范围，然后打开该范围内的端口。

客户端和FTP Server之间的连接经常断开

- 问题原因

客户端和FTP Server之间连接超时。

- 解决方法

设置客户端和FTP Server之间连接不超时。以FileZilla工具为例，在设置 > 连接，将超时设置为0（表示不超时）。

7.osscli (已下线)

7.1. 快速安装

osscli是基于 Python 2.x 的命令行工具，支持Bucket管理、文件管理等功能。

 **说明** osscli操作命令已整合到ossutil中，工具已于2019年7月31日下线，给您带来不便敬请谅解。

使用场景

osscli适用于以下场景：

- API级别的开发、调试，比如发送特定格式的请求、分步骤执行分片上传等。
- Bucket配置，不方便使用控制台情况下的Bucket配置，如logging/website/lifecycle等。

使用限制

- osscli支持的运行环境包括Python 2.5/2.6/2.7，不支持Python 3.x。
- 在Python SDK 0.x基础上开发，Python SDK 0.x已经不再维护，目前维护的Python SDK是2.x.x。
- osscli只进行BUG修改，不再支持新功能，如低频存储/归档存储、跨区域复制、镜像回源等。

安装使用

对下载的Python SDK压缩包进行解压后，在osscli所在目录直接执行 `python osscli + 操作` 即可。比如上传一个文件到Bucket：

```
python osscli put myfile.txt oss://mybucket
```

 **说明** osscli中用oss://bucket或者oss://bucket/object表示这是一个Bucket或者Object。oss://只是一种资源的表示方式，没有其他意义。

如果需要详细的命令列表输入：`python osscli`。

如果需要详细的参数列表说明输入：`python osscli help`。

7.2. 使用示例

本文介绍osscli的使用示例。

 **注意** osscli操作命令已整合到ossutil中，工具已于2019年7月31日下线，给您带来的不便敬请谅解。

安装配置osscli

在Linux或者Windows上下载SDK安装包后，解压缩后即可使用osscli。

使用osscli时直接调用Python osscli即可获取相应的说明。每种命令有两种执行模式。以执行gs命令查询用户所创建的Bucket为例：

- 方法1：不指定ID和KEY，osscli从默认文件中读取ID和KEY。

```
$ python osscli gs
can't get accessid/accesskey, setup use : config --id=accessid --key=accesskey
```

❓ 说明 如果出现这样的提示，表明没有配置好ID和KEY，见方法2中提示的配置命令。

如果配置好ID和KEY，并且ID和KEY有效，执行

```
$ python osscli gs
2021-07-19 08:11 test-oss-sample
Bucket Number is: 1
```

- 方法2：直接在命令中指定ID和KEY，osscli从命令行中读取ID和KEY。如果ID和KEY有效，执行后得到以下结果。

```
$ python osscli gs --id=your_id --key=your_key --host=your_endpoint
2021-07-19 08:11 test-oss-sample
Bucket Number is: 1
```

如果要配置用户的ID和KEY到默认的文件中，请运行如下命令用来配置访问OSS所需要的ID和KEY。默认的OSS HOST为oss.aliyuncs.com。

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

如果出现类似 “Your configuration is saved into ” 的提示，表明ID和KEY已经保存成功。

基础操作

- 列出创建的Bucket

```
$python osscli getallbucket
```

如果是初次使用OSS的用户且没有创建Bucket，则输出为空。

- 创建Bucket

创建一个名为mybucketname的Bucket。

```
$python osscli createbucket mybucketname
```

创建该Bucket可能不成功。原因是OSS中的Bucket名字是全局唯一的，此时您需要换一个Bucket名字，如在Bucket名字中加入特定的日期。

- 查看是否创建成功

```
$python osscli getallbucket
```

如果没有成功请检查osscli返回的错误信息。

- 查看Object

成功创建Bucket后，查看Bucket中有哪些Object。

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

由于Bucket中还没有object，输出为空。

- 上传object

向Bucket中上传一个Object。假如本地文件名叫local_existed_file，其MD5值如下所示。

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

 说明 `md5sum` 为 Linux 命令，Windows下无此命令。

- 再次查看Object

如果创建成功，再次查看Bucket中有哪些Object。

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

- 下载Object

从Bucket中下载Object到本地文件，并比对下载的文件MD5值

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

 说明 `md5sum` 为 Linux 命令，Windows下无此命令。

- 删除Object

```
$ python osscli delete oss://mybucketname/test_object
```

- 删除Bucket

 说明 如果Bucket中还有Object，则这个Bucket不能被删除。

```
$ python osscli deletebucket mybucketname
```

使用lifecycle

- 配置一个lifecycle的xml格式的文本文件

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

表示删除Bucket下以log_backup/ 为前缀，并且相对当前时间超过2天的Object。详细的规则配置可以参考[API文档](#)

- 写入lifecycle

```
python osscli putlifecycle oss://mybucket lifecycle.xml
0.150(s) elapsed
```

- 读取lifecycle

```
python osscli getlifecycle oss://mybucket
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>1125</ID>
    <Prefix>log_backup</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>2</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
0.027(s) elapsed
```

- 删除lifecycle

```
python osscli deletelifecycle oss://mybucket
0.139(s) elapsed
```

- 读取lifecycle

```
python osscli 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!
```

防盗链设置

- 允许空referer访问

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

- 获取referer配置

```
$osscli getreferer oss://test
<?xml version="1.0" encoding="UTF-8"?>
<RefererConfiguration>
  <AllowEmptyReferer>true</AllowEmptyReferer>
  <RefererList />
</RefererConfiguration>
```

- 不允许空referer, 只允许referer为www.example.com的请求

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

- 获取referer配置

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

- 不允许空referer, 只允许referer为www.example.com和www.example.org的请求

```
$osscli putreferer oss://test --allow_empty_referer=false --referer='www.example.com,www.
example.org'
```

- 获取设置的referer

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

使用logging

- 设置logging

```
$osscli putlogging oss://mybucket oss://myloggingbucket/mb
```

- 获取设置的logging

```
$osscli getlogging oss://mybucket
```

7.3. 有关Bucket命令

本文主要介绍与存储空间（Bucket）相关的命令。

 **注意** osscli操作命令已整合到ossutil中，工具已于2019年7月31日下线，给您带来不便敬请谅解。

config

命令说明：

```
config --id=[accessid] --key=[accesskey] --host=[host] --sts_token=[sts_token]
```

使用示范：

- `python osscli config --id=your_id --key=your_key`

```
python osscli config --id=your_id --key=your_key
--host=oss-internal.aliyuncs.com
```

getallbucket(gs)

命令说明：

```
getallbucket(gs)
```

获取创建的bucket。gs是get allbucket的简写。gs和get allbucket是同样的效果。

使用示范：

- `python osscli getallbucket`
- `python osscli gs`

createbucket(cb,mb,pb)

命令说明：

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

创建bucket的命令。

- cb是create bucket的简写、mb是make bucket的简写、pb是put bucket的简写。
- oss://bucket表示bucket。
- acl参数可以传入，也可以不传入。

使用示范：

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

deletebucket(db)

命令说明：

```
deletebucket(db) oss://bucket
```

删除bucket的命令，db是delete bucket的简写。

使用示范：

- `python osscli deletebucket oss://mybucket`
- `python osscli db oss://myfirstbucket`

deletewholebucket

 **警告** 该命令将会删除所有的数据，且不可恢复。请慎重使用。

命令说明：

```
deletewholebucket oss://bucket
```

删除bucket及其内部object以及multipart相关的内容。

使用示范：

```
python osscli deletewholebucket oss://mybucket
```

getacl

命令说明：

```
getacl oss://bucket
```

获取bucket的访问控制权限。

使用示范：

```
python osscli getacl oss://mybucket
```

setacl

命令说明：

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

修改bucket的访问控制权限。acl允许设置的访问控制权限包括private、public-read、public-read-write。

使用示范：

```
python osscli setacl oss://mybucket --acl=private
```

putlifecycle

命令说明：

```
putlifecycle oss://mybucket lifecycle.xml
```

设置lifecycle规则。其中lifecycle.xml为XML格式的lifecycle配置文件，详细的规则配置可以参考[API文档](#)。

使用示范：

```
python osscli putlifecycle oss://mybucket lifecycle.xml
```

示例：

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

getlifecycle

命令说明：

```
osscli getlifecycle oss://bucket
```

获取该Bucket lifecycle规则。

使用示范：

```
python osscli getlifecycle oss://mybucket
```

deletelifecycle

命令说明：

```
osscli deletelifecycle oss://bucket
```

删除该bucket下所有的lifecycle规则。

使用示范：

```
python osscli deletelifecycle oss://mybucket
```

putreferer

命令说明：

```
osscli putreferer oss://bucket --allow_empty_referer=[true|false]
--referer=[referer]
```

设置防盗链规则。其中参数 `allow_empty_referer` 用来设置是否允许为空，为必选参数。参数 `referer` 用来设置允许访问的白名单，例如 “www.example.com,www.example.org”，以 “,” 作为分隔。详细的配置规则参考[产品文档](#)。

使用示范：

```
python osscli putreferer oss://mybucket --allow_empty_referer=true
--referer="www.example.com,www.example.org"
```

getreferer

命令说明：

```
osscli getreferer oss://bucket
```

获取该Bucket下防盗链设置规则。

使用示范：

```
python osscli getreferer oss://mybucket
```

putlogging

命令说明：

```
osscli putlogging oss://source_bucket oss://target_bucket/[prefix]
```

其中source_bucket表示需要记录日志的bucket，而target_bucket则是用来存放产生的日志。允许对源bucket产生的日志文件设置前缀，方便用户归类查询。

使用示范：

```
python osscli getlogging oss://mybucket
```

getlogging

命令说明：

```
osscli getlogging oss://bucket
```

获取该bucket的logging设置规则。

使用示范：

```
python osscli getlogging oss://mybucket
```

7.4. 有关Object命令

本文主要介绍与对象（Object）相关的命令。



注意 osscli操作命令已整合到ossutil中，工具已于2019年7月31日下线，给您带来不便敬请谅解。

ls(list)

命令说明：

```
ls(list) oss://bucket/[prefix] [marker] [delimiter] [maxkeys]
```

列出存储空间（Bucket）中的Object。填写prefix时，可以列出以指定前缀开头的所有文件。例如，prefix为abc，则列出名称以abc开头的所有Object。

使用示范：

- ```
python osscli ls oss://mybucket/folder1/folder2
```
- ```
python osscli ls oss://mybucket/folder1/folder2 marker1
```
- ```
python osscli ls oss://mybucket/folder1/folder2 marker1 /
```
- ```
python osscli ls oss://mybucket/
```
- ```
python osscli list oss://mybucket/ "" "" 100
```

命令说明：

```
ls(list) oss://bucket/[prefix] --marker=xxx --delimiter=xxx --maxkeys=xxx
--encoding_type=url
```

列出Bucket中的Object。其中encoding\_type可以指定传输中使用的编码。当指定为URL编码时，支持显示含控制字符的Object。

使用示范：

- `python osscli ls oss://mybucket/folder1/folder2 --delimiter=/`
- `python osscli ls oss://mybucket/folder1/folder2 --marker=a`
- `python osscli ls oss://mybucket/folder1/folder2 --maxkeys=10`

## mkdir

命令说明：

```
mkdir oss://bucket/dirname
```

创建一个文件夹。

使用示范：

```
python osscli mkdir oss://mybucket/folder
```

## listallobject

命令说明：

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

列出Bucket下所有的Object，可以指定prefix来显示。

使用示范：

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

## deleteallobject

命令说明：

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

删除Bucket下所有的Object，可以指定prefix来删除。

使用示范：

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

## downloadallobject

命令说明：

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

将Bucket下的Object下载到本地目录，并且保持目录结构。可以指定prefix下载。—replace=false表示下载时不会覆盖本地的同名文件，为true时则覆盖。同时可以通过thread\_num来配置下载线程。

使用示范：

- `python osscli downloadallobject oss://mybucket /tmp/folder`

```
python osscli downloadallobject oss://mybucket /tmp/folder
--replace=false
```

```
python osscli downloadallobject oss://mybucket /tmp/folder --replace=true
--thread_num=5
```

## downloadtodir

命令说明：

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

将Bucket下的Object下载到本地目录，并且保持目录结构。可以指定prefix下载。--replace=false表示下载时不会覆盖本地的同名文件，为true时则覆盖。downloadtodir与downloadallobject 效果一样。

使用示范：

- ```
python osscli downloadtodir oss://mybucket /tmp/folder
```
- ```
python osscli downloadtodir oss://mybucket /tmp/folder --replace=false
```

```
python osscli downloadtodir oss://mybucket /tmp/folder
--replace=true
```

## uploadfromdir

命令说明：

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

将本地目录里的文件上传到Bucket中。

例如localdir为 /tmp/，里面有a/b、a/c、a三个文件，则上传到OSS中为oss://bucket/a/b、oss://bucket/a/c、oss://bucket/a。如果指定了prefix为mytest，则上传到OSS中为oss://bucket/mytest/a/b、oss://bucket/mytest/a/c、oss://bucket/mytest/a。

--check\_point=check\_point\_file 用于指定文件。指定文件后，osscli会将已经上传的本地文件以时间戳的方式存放到check\_point\_file中，uploadfromdir命令会将正在上传的文件的时间戳和check\_point\_file记录的时间戳进行比较。如果有变化则会重新上传，否则跳过。默认情况下没有check\_point\_file。--replace=false表示下载时不会覆盖本地的同名文件，为true时则覆盖。--check\_md5=false 表示上传文件时不会校验携带Content-MD5请求头，为true时则校验。

注意：check\_point\_file文件中记录的是上传的所有文件的。

使用示范：

- ```
python osscli uploadfromdir /mytemp/folder oss://mybucket
```

```
python osscli uploadfromdir /mytemp/folder oss://mybucket
--check_point_file=tmp/mytemp_record.txt
```

```
python osscli uploadfromdir C:\Documents and Settings\User\My Documents\Downloads
oss://mybucket --check_point_file=C:\cp.txt
```

put

命令说明：

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

上传一个本地的文件到Bucket中，可以指定Object的content-type，或指定自定义的headers。 `--check_md5=false` 表示上传文件时不会校验携带Content-MD5请求头，为true时则校验。

使用示范：

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

python osscli put myfile.txt oss://mybucket/test.txt
--content-type=plain/text
```

## upload

### 命令说明：

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

将本地文件以Object group的形式上传。不推荐使用。 `--check_md5=false` 表示上传文件时不会校验携带Content-MD5请求头，为true时则校验。

### 使用示范：

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

## get

### 命令说明：

```
get oss://bucket/object localfile
```

将object下载到本地文件。

### 使用示范：

```
python osscli get oss://mybucket/myobject /tmp/localfile
```

## multiget(multi\_get)

### 命令说明：

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

将Object以多线程的方式下载到本地文件。同时可以配置线程数。

### 使用示范：

- `python osscli multiget oss://mybucket/myobject /tmp/localfile`

- ```
python osscli multi_get oss://mybucket/myobject /tmp/localfile
```

cat

命令说明：

```
cat oss://bucket/object
```

读取Object的内容，直接打印出来。在Object内容比较大的时候请不要使用。

使用示范：

```
python osscli cat oss://mybucket/myobject
```

meta

命令说明：

```
meta oss://bucket/object
```

读取Object的meta信息，打印出来。meta信息包括content-type、文件长度、自定义meta等内容。

使用示范：

```
python osscli meta oss://mybucket/myobject
```

copy

命令说明：

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

将源Bucket中的源Object复制到目的Bucket中的目的Object。

使用示范：

```
python osscli copy oss://bucket1/object1 oss://bucket2/object2
```

rm(delete,del)

命令说明：

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

删除Object。当指定encoding-type为URL编码时，传入待删除的字串也需为URL编码。

使用示范：

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

signurl(sign)

命令说明：

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

生成包含签名的URL，并指定超时时间。适用于bucket为私有时将特定的Object提供他人访问。

使用示范：

- `python osscli sign oss://mybucket/myobject`
- `python osscli signurl oss://mybucket/myobject`

7.5. 有关Multipart命令

本文主要介绍与碎片（part）相关的命令。



注意 osscli操作命令已整合到ossutil中，工具已于2019年7月31日下线，给您带来不便敬请谅解。

init

命令说明：

```
init oss://bucket/object
```

初始化生成一个Upload ID。这个Upload ID可以配合后面的multiupload命令来使用。

使用示例：

```
python osscli init oss://mybucket/myobject
```

listpart

命令说明：

```
listpart oss://bucket/object --upload_id=xxx
```

显示指定object的Upload ID下已经上传的Parts。相关概念见[OSS API文档](#)。必须要指定Upload ID。

使用示例：

```
python osscli listpart oss://mybucket/myobject --upload_id=
75835E389EA648C0B93571B6A46023F3
```

listparts

命令说明：

```
listparts oss://bucket
```

显示bucket中未完成的multipart Upload ID和object。一般在删除bucket提示bucket非空的情况下，可以用这个命令查看是否有multipart相关的内容。

使用示例：

```
python osscli listparts oss://mybucket
```

getallpartsize

命令说明：

```
getallpartsize oss://bucket
```

显示bucket中还存在的Upload ID下已经上传的Parts的总大小。

使用示例：


```
python osscli getallpartsize oss://mybucket
```

cancel

命令说明:

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

终止Upload ID对应的Multipart Upload事件。

使用示例:

```
python osscli cancel oss://mybucket/myobject --upload_id=
D9D278DB6F8845E9AFE797DD235DC576
```

multiupload(multi_upload,mp)

命令说明:

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

将本地文件以multipart的方式上传到OSS。

使用示例:

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

命令说明:

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

将本地文件以multipart的方式上传到OSS。本地文件划分的块数由max\_part\_num来指定。这个命令在实现的时候，会先去判断Upload ID对应的Parts的ETag是否和本地文件的MD5值是否相等，相等则跳过上传。所以在使用之前生成一个Upload ID，作为参数传进来。即使上传没有成功，重复执行相同的multiupload命令可以达到一个断点续传的效果。`--check_md5=false` 表示上传文件时，不会做携带Content-MD5请求头校验。true则会做校验。

使用示例:

```
python osscli multiupload /tmp/localfile.txt oss://mybucket/object --upload_id=
D9D278DB6F8845E9AFE797DD235DC576
```

```
python osscli multiup_load /tmp/localfile.txt oss://mybucket/object
--thread_num=5
```

- ```
python osscli mp /tmp/localfile.txt oss://mybucket/object --max_part_num=100
```

copylargefile

命令说明:

```
copylargefile oss://source_bucket/source_object oss://target_bucket/target_object
--part_size=10*1024*1024 --upload_id=xxx
```

对于超过1G的大文件进行复制时，采用multipart的方式将object复制到指定位置（源bucket必须与目标bucket处于同一region）。其中upload_id为可选参数，当需要对某一次multipart copy事件进行续传的时候，可以传入该事件的upload_id。part_size用来设定分块大小，分块最小需要大于100KB，最多支持10000块分块。如果part_size设定值导致与OSS限制冲突，程序会帮你自动调节分块大小。

使用示例：

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

uploadpartfromfile (upff)

命令说明：

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

主要用于测试，不推荐使用。

uploadpartfromstring(upfs)

命令说明：

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

主要用于测试，不推荐使用。

8. RAM策略编辑器

本文介绍RAM策略编辑器的使用方法。

地址

[RAM Policy Editor](#)

使用

RAM授权策略由若干条规则组成。使用RAM策略编辑器，可以在界面上逐条添加或删除规则，并自动生成策略的JSON文本。添加所有规则后，只需要将JSON文本拷贝并粘贴到访问控制（RAM）控制台的创建授权策略内容框内即可使用。具体操作，请参见[创建自定义权限策略](#)。

RAM策略编辑器中，每条规则需要设置其Effect、Actions、Resources和Conditions：

- Effect

指定这条规则是允许访问（Allow）还是禁止访问（Deny）。

- Actions

指定访问资源的动作，可以选择多项。一般来说用户使用提供的通配动作就足够了：

- `oss:*` 表示允许所有动作。
- `oss:Get*` 表示允许所有的读动作。
- `oss:Put*` 表示允许所有的写动作。

更多信息，请参见[RAM Policy Editor README](#)。

- Resources

指定授权访问的OSS的资源，可以指定多个。Resources常见场景如下：

- 表示某个Bucket：`my-bucket`（此时对bucket下的文件没有权限）
- 表示某个Bucket下面所有文件：`my-bucket/*`（此时对bucket本身没有权限，例如List Objects）
- 表示某个Bucket下某个目录：`my-bucket/dir`（此时对dir/下面的文件没有权限）
- 表示某个Bucket下某个目录下面所有文件：`my-bucket/dir/*`（此时对dir没有权限，例如List Objects）
- 填写完整的资源路径：`acs:oss:*:174649585760xxxx:my-bucket/dir`，其中 `174649585760xxxx` 为用户的UID（可在控制台查看）

- EnablePath

当您需要对某个目录授权时，往往还需要保证对上一层目录也有List权限，例如对 `my-bucket/users/dir/*` 赋予读写权限，为了在控制台（或其他工具）能够查看这个目录，还需要以下权限：

```
ListObjects my-bucket
ListObjects my-bucket/users
ListObjects my-bucket/users/dir
```

勾选EnablePath选项时，会自动添加以上权限。

- Conditions

指定授权访问时应该满足的条件，可以指定多个。

示例

授权对 my-bucket 及其文件所有权限示例如下：

RAM Policy Editor v1.0.4

Star 2

添加规则

Effect

Allow

Actions

oss:*

Resources

my-bucket

my-bucket/*

每添加一个Resource后按回车确认

例子: my-bucket, my-bucket/dir/

More...

EnablePath

☐ 自动设置父目录权限 ?

Conditions (Optional)

显示

添加规则

授权策略

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "oss:*"
      ],
      "Resource": [
        "acs:oss:*:*:my-bucket",
        "acs:oss:*:*:my-bucket/*"
      ],
      "Condition": {}
    }
  ]
}
```

规则列表

Effect	Actions	Resources	Conditions
Allow	oss:*	acs:oss:*:*:my-bucket acs:oss:*:*:my-bucket/*	