

阿里云 对象存储

常用工具

文档版本：20190919

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
	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。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 确定 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
##	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[]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>swich {stand slave}</code>


目录


法律声明.....	I
通用约定.....	I
1 OSS常用工具汇总.....	1
2 图形化管理工具ossbrowser.....	4
2.1 快速开始.....	4
2.2 权限管理.....	8
2.3 常见问题.....	12
3 命令行工具ossutil.....	15
3.1 概述.....	15
3.2 下载和安装.....	16
3.3 常用命令.....	18
3.3.1 appendfromfile.....	18
3.3.2 bucket-encryption.....	20
3.3.3 bucket-policy.....	22
3.3.4 bucket-tagging.....	24
3.3.5 cat.....	26
3.3.6 config.....	27
3.3.7 cors.....	30
3.3.8 cors-options.....	31
3.3.9 cp.....	34
3.3.10 create-symlink.....	46
3.3.11 du.....	47
3.3.12 getallpartsize.....	49
3.3.13 hash.....	50
3.3.14 help.....	51
3.3.15 lifecycle.....	51
3.3.16 listpart.....	54
3.3.17 logging.....	54
3.3.18 ls.....	56
3.3.19 mb.....	60
3.3.20 mkdir.....	62
3.3.21 object-tagging.....	63
3.3.22 probe.....	65
3.3.23 read-symlink.....	68
3.3.24 referer.....	70
3.3.25 restore.....	72
3.3.26 request-payment.....	73
3.3.27 rm.....	75
3.3.28 set-acl.....	78
3.3.29 set-meta.....	80



3.3.30 sign.....	82
3.3.31 stat.....	84
3.3.32 update.....	85
3.3.33 website.....	86
3.4 查看选项.....	89
3.5 常见问题.....	95
4 数据迁移工具ossimport.....	97
4.1 说明及配置.....	97
4.2 单机部署.....	109
4.3 分布式部署.....	113
4.4 常见问题.....	117
5 RAM策略编辑器.....	125
6 ossftp.....	128
6.1 如何快速安装ossftp.....	128
6.2 Discuz如何存储远程附件到OSS.....	131
6.3 Phpwind如何存储远程附件到OSS.....	135
6.4 Wordpress如何存储远程附件到OSS.....	139
6.5 如何结合RAM实现文件共享.....	143
6.6 常见问题.....	150
7 ossfs.....	152
7.1 快速安装.....	152
7.2 常见问题.....	155
7.3 ossfs使用问题排查案例.....	161
8 osscmd（已下线）.....	165
8.1 快速安装.....	165
8.2 使用示例.....	166
8.3 有关Bucket命令.....	170
8.4 有关Object命令.....	175
8.5 有关Multipart命令.....	180
9 通过数据集成导入数据.....	184

1 OSS常用工具汇总

除控制台外，您还可以使用以下常用工具帮助您更高效地使用OSS。

工具	简介
ossbrowser	<p>图形化的Object管理工具。</p> <ul style="list-style-type: none">· 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子账号的图形化Policy授权操作。· 支持Windows、Linux、Mac平台。 <p>使用限制：</p> <ul style="list-style-type: none">· ossbrowser是图形化工具，传输速度和性能不如ossutil。· 只支持5GB以下的文件上传或复制。
ossutil	<p>Object和Bucket的命令行管理工具。</p> <ul style="list-style-type: none">· 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。· 支持Bucket常见管理命令。
osscmd（已下线）	<p>Object和Bucket的命令行管理工具。</p> <ul style="list-style-type: none">· 提供完备的Bucket、Object管理命令。· 支持Windows、Linux平台。 <p>使用限制：</p> <ul style="list-style-type: none">· 仅适用于Python2.5 - 2.7版本，不支持Python 3.x版本。· 不再支持OSS新增功能，如低频存储/归档存储、跨区域复制、镜像回源等。 <div> 注意： osscmd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。</div>

工具	简介
ossfs	<p>Bucket挂载工具。</p> <p>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。</p> <ul style="list-style-type: none">· 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。 <p>使用限制：</p> <ul style="list-style-type: none">· 不支持挂载归档型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list directory，因为需要远程访问OSS服务器，所以性能较差。· 重命名文件/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。例如，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。 <div> 说明： 相比于ossfs，建议您优先使用阿里云存储网关产品，详情请参见云存储网关。</div>
ossftp	<p>管理Object的FTP工具。</p> <ul style="list-style-type: none">· 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，将对文件、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。

工具	简介
ossimport	<p>OSS数据同步工具：</p> <ul style="list-style-type: none">· 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及以上版本。 <div> 说明： 您也可以使用阿里云数据在线迁移服务，无需部署迁移工具。</div>
可视化签名工具	<p>可视化签名工具（第三方工具）。</p> <ul style="list-style-type: none">· 可用于生成数据的签名URL。· 可调试签名中遇到的问题。当您在实现OSS的签名过程中遇到问题，请与该工具的签名对比，以便快速定位该错误。· 网页版支持浏览器Chrome、Firefox、Safari。
RAM Policy Editor	<p>OSS授权策略自动化生产工具。</p> <ul style="list-style-type: none">· 可根据填写的需求自动生成授权策略，您可将自动生成的策略移植到RAM访问控制的自定义权限内。· 支持浏览器Chrome、Firefox、Safari。 <div> 说明： 当您需要生成自定义授权策略时，推荐使用该工具。</div>
oss-emulator	<p>轻量级的OSS服务模拟器。</p> <ul style="list-style-type: none">· 提供与OSS服务相同的API接口，适用于基于OSS应用的调试、性能测试等。· 基于Ruby 2.2.8及以上版本。· 支持Windows、Linux平台。

2 图形化管理工具ossbrowser

2.1 快速开始

ossbrowser 是 OSS 官方提供的图形化管理工具，提供类似 Windows 资源管理器的功能。使用 ossbrowser，您可以方便地浏览、上传、下载和管理文件。

注意事项

- 只支持 5GB 以下文件的移动或复制，5GB 以上文件的移动或复制操作建议使用 [ossutil](#)。
- 支持 Windows7 及以上版本、Linux 和 Mac 平台，不建议在 Windows XP 和 Windows Server 平台上使用。

快速安装

1. 下载并安装 ossbrowser。

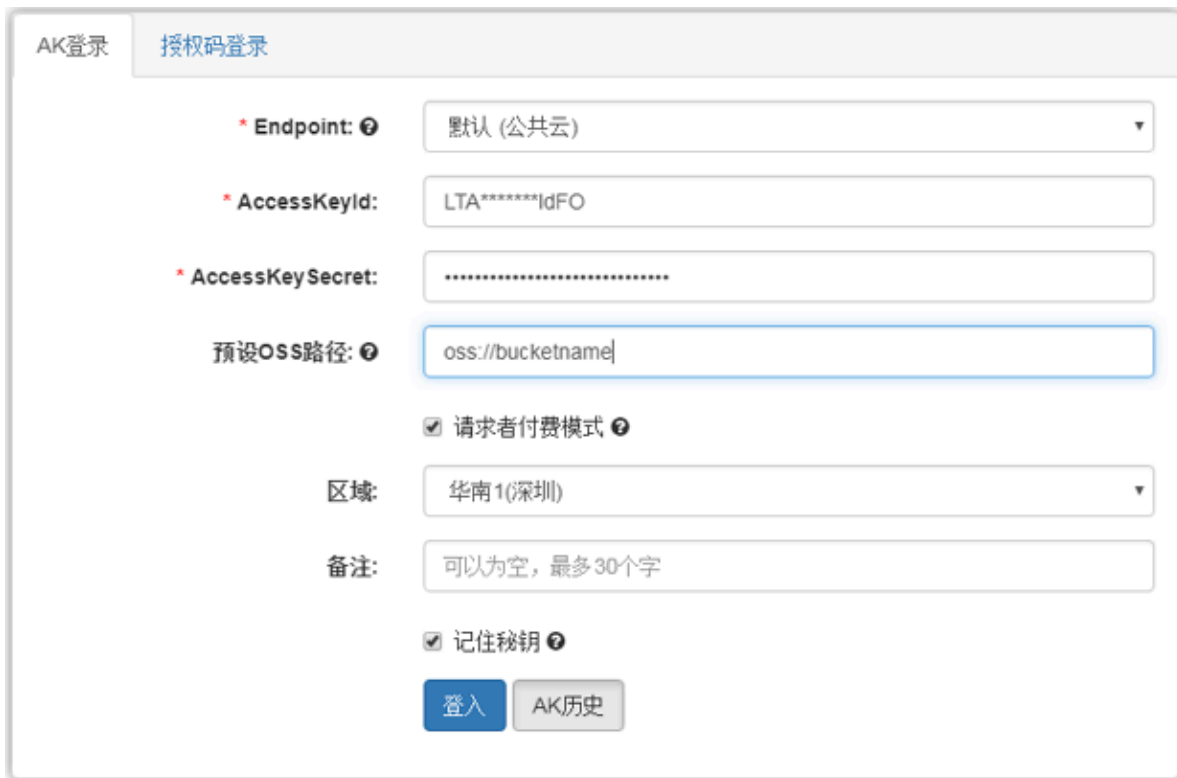
支持平台	下载地址
Windows x32	Windows x32
Windows x64	Windows x64
MAC	MAC
Linux x64	Linux x64



说明:

更多下载地址请参见 [GitHub](#)。

2. 打开 ossbrowser 并登录。



- Endpoint：选择登录的地域域名。
 - 默认：使用默认的 Endpoint 登录。
 - 自定义：输入 Endpoint 信息，填写 http 或 https 加对应地域的域名，选择使用 http 或 https 方式登录，例如 https://oss-cn-beijing.aliyuncs.com。地域域名信息请参见[#unique_15](#)。
 - cname：可使用绑定的自定义域名登录。绑定自定义域名请参见[#unique_16](#)。
- AccessKeyId/AccessKeySecret：填写账号的 AccessKey（AK）信息。为保证数据安全，推荐您使用 RAM 账号（子账号）的 AK 登录 ossbrowser。获取 AK 信息请参考[#unique_17](#)。
- 预设 OSS 路径：
 - 拥有所有 Bucket 管理权限的管理员子账号：无需配置。
 - 操作员子账号：需要配置。填写您需要访问的 OSS Bucket 名称或子目录（该 RAM 子账号需要有该路径的访问权限），格式为 `oss://bucketname/path`。
 - 请求者付费模式：如果对应的路径开启了请求者付费模式，需勾选此项。请求者付费模式介绍请参见[#unique_18](#)。
 - 区域：选择 OSS 路径对应的地域。
- 记住密钥：勾选可保存 AK 密钥。再次登录时，单击 AK 历史，可选择该密钥登录，不需要手动输入 AK，请不要在临时使用的电脑上勾选。

管理 Bucket

· 创建 Bucket

1. 在 ossbrowser 首页，单击新建 Bucket。
2. 配置 Bucket 信息。
 - 名称：最大 63 字符，名称需唯一，不可重名。
 - 区域：选择 Bucket 所在地域。
 - ACL 权限：选择 Bucket 的 ACL 权限。ACL 权限介绍请参见[基于读写权限 ACL 的权限控制](#)。
 - 类型：选择 Bucket 的默认存储类型。存储类型详情请参见[#unique_20](#)。
3. 配置完成后单击确定。

· 删除 Bucket

选中指定的 Bucket，之后单击更多 > 删除即可。当 Bucket 内存在文件或碎片时，无法被删除。

管理 Object

· 创建目录

1. 在 ossbrowser 首页，单击对应的 Bucket。
2. 单击创建目录。
3. 填写目录名并单击确定。



说明:

- 不允许使用表情符，请使用符合要求的 UTF-8 字符。
- 一次仅可以创建一个目录，且仅可以创建单级目录，无法创建多级目录。例如：可以创建目录 `abc`，无法创建目录 `abc/123`。
- 不允许出现名为 `..` 的子目录。
- 总长度控制在 1-254 个字符。

· 上传文件/目录

在指定的 Bucket 或目录内，单击上传/目录，之后选择需要上传的文件/目录，即可完成上传文件/目录的操作。上传时可同时选择多个数据进行批量上传。

· 下载文件/目录

在指定的 Bucket 或目录内，选中需要下载的数据，单击下载即可。可同时选中多个数据进行批量下载。

- 复制文件/目录

1. 在指定的 Bucket 或目录内，选中需要复制的数据，单击复制。
2. 进入需要粘贴文件的 Bucket 或目录，单击粘贴即可。若复制文件的源地址和目的地址一致，则会覆盖原有文件。若该文件是低频或归档类型文件，且存储天数未满足指定天数，会产生提前删除费用。提前删除费用请参考[计量项和计费项](#)。

- 移动文件/目录

1. 在指定的 Bucket 或目录内，选中需要移动的文件/目录，单击更多 > 移动。
2. 进入需要粘贴文件的 Bucket 或目录，单击粘贴即可。



注意：

移动文件/目录，实际上是将文件/目录从源地址复制到目的地址，之后将源地址的文件删除。若被移动的文件是低频或归档类型文件，且存储天数未满足指定天数，会产生提前删除费用。

- 重命名文件/目录

在指定的 Bucket 或目录内，选中需要重命名的文件/目录，单击更多 > 重命名，输入新的名称即可。



注意：

- 重命名操作仅针对 1GB 以下的文件有效。
- 重命名文件/目录，实际上是将文件/目录复制后重新命名，并保存，之后将原文件删除。若被重命名的文件/目录是低频或归档类型文件，且存储天数未满足指定天数，会产生提前删除费用。

- 删除文件/目录

选择需要删除的文件/目录，单击更多 > 删除。若被删除的文件/目录是低频或归档类型文件，且存储天数未满足指定天数，会产生提前删除费用。

- 生成文件 URL

1. 选择指定文件，单击更多 > 获取地址
2. 填写链接有效时间，并单击生成。
3. 单击点击复制或发送邮件，将文件 URL 发送给需要访问该文件的用户。扫描生成的二维码也可以访问该文件。

- 预览文件

双击指定的文件即可完成预览。ossbrowser 目前支持 txt、pdf 和大部分图片文件的预览。

- 管理碎片

选择指定的 Bucket，之后单击碎片。通过管理碎片，您可以将不需要的碎片删除。

2.2 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 子账号

为保证数据安全，推荐您使用 RAM 账号（子账号）的 AccessKey（AK）登录 ossbrowser。



说明：

创建子账号和 AK。详情请参见[创建RAM用户](#)。

子帐号的权限分为：

- 管理员子账号：拥有管理权限的子账号。例如授予某个子账号可管理所有Bucket和具有 RAM 授权配置的权限，该子账号即为管理员子账号。您可以使用阿里云主账户登录RAM控制台，创建管理员子账号，对账号授予如下权限。

编辑个人授权策略

添加授权策略后，该账户即具有该条策略的权限，同一条授权策略不能被重复添加。

可选授权策略名称	类型
请输入关键词查询	
AliyunOSSReadOnlyAccess 只读访问对象存储服务(OSS)的权限	系统
AliyunECSFullAccess 管理云服务器服务(ECS)的权限	系统
AliyunECSReadOnlyAccess 只读访问云服务器服务(ECS)的权限	系统
AliyunRDSFullAccess 管理云数据库服务(RDS)的权限	系统

已选授权策略名称	类型
AliyunOSSFullAccess 管理对象存储服务(OSS)权限	系统
AliyunRAMFullAccess 管理访问控制(RAM)的权限，即管理...	系统
AliyunSTSAssumeRoleAccess 调用STS服务AssumeRole接...	系统

确定 关闭

- 操作员子账号：仅拥有某个 Bucket 或某个目录只读权限的子账号。管理员可通过[简化 Policy 授权](#)给予子账号分配权限。



说明：

您可以为子帐号授予更小的权限，具体设置请参考[权限管理概述](#)。

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。

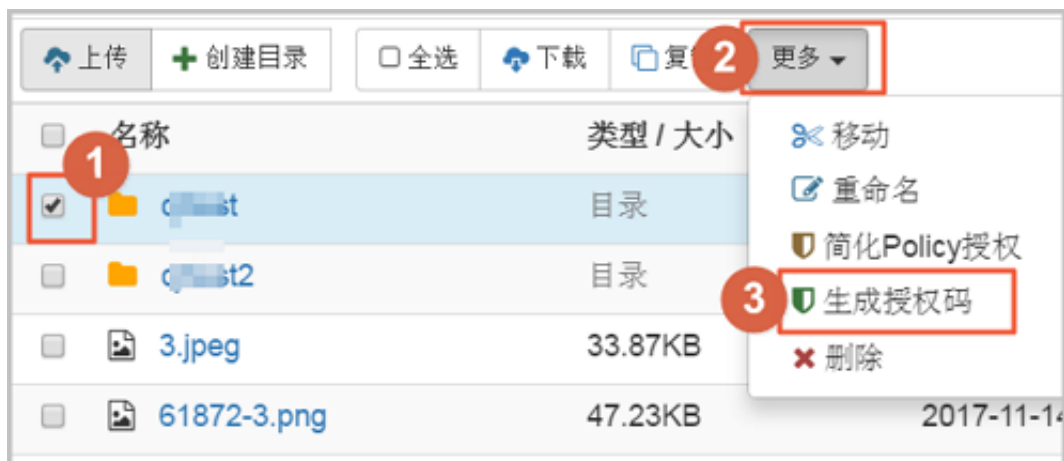
1. 使用管理员子账号登录 ossbrowser。



注意：

为了您的数据安全，当您使用主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。



AK登录 授权码登录

* 授权码: ?

0TFJBai9OZzNEM3ZITfInMTIBRTZ2L3ZqdEVYT3FMKzVDQ3V3aG1
VbDg...0FFa0Z
qMHk...XBrC0pJ
VGllc...MnU5dF
ZIOVc...VJidW5o
VGxIT...45OUo4
bHRq...joiMjAxN
y0xM...taGFuZ3
pob3UiLCJvc3NwYXRoIjoib3NzOi8vZ3VhbmdjaHVuL2RibW8vliwicH
JpdmlsZWdlIjoicmVhZE9ubHkifQ==

(有效期至2017-10-27 19:17:02, 剩余时间: 58 分 28 秒)

登入

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。



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，对子账号进行更细致的管理。

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。

简化Policy授权

将下列资源:

1. Koala.jpg

的权限:

☒ 只读权限 ☐ 读写权限 **1**

[\[查看Policy\]](#) **2**

创建为policy, 命名为:

plc-readOnly-Koala-jpg

简化Policy授权 ✕

将下列资源:

1. Koala.jpg

的权限:

☒ 只读权限 ☐ 读写权限

[\[查看Policy\]](#)

创建为policy, 命名为:

plc-readOnly-Koala-jpg

并授权给:

[新建一个] ▼

新用户名: 随机生成

AK发送到邮件:

还没设置邮件发送配置, 需要先设置 [打开设置](#)

✓ 确定授权 ✕ 关闭

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 子账号、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。

2.3 常见问题

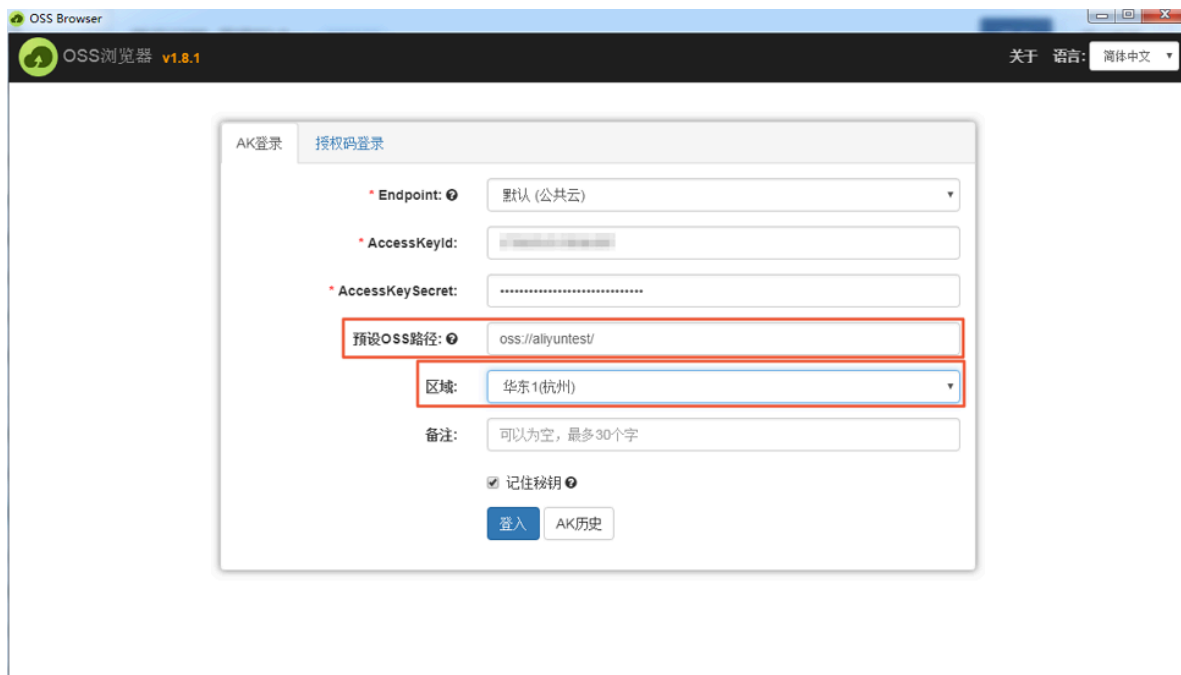
本文介绍在使用 ossbrowser 时遇到的一些常见问题及解决方法。

使用 AK 登录时出现 “AccessDenied:You are forbidden to list buckets.” 报错

问题分析：登录时出现该报错，表明您登录的账号无所有 Bucket 的访问权限。

解决方法：

- 若您的账号无 Bucket 访问权限，请先添加访问权限，之后再登录。
- 若您的账号仅有部分 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。例如：某账号仅有名为 *aliyuntest*，位于“华东1（杭州）”的 Bucket 访问权限，则预设 OSS 路径应填写：*oss://aliyuntest/*，区域需选择“华东1（杭州）”。

**注意:**

若账号拥有多个 Bucket 但是没有所有 Bucket 访问权限的时候，需分次登录管理对应的 Bucket。

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/下载任务，剩余的任务会在队列中等待前面的任务完成后再次执行，在文件较多的情况下，默认的设置会影响到上传/下载的速度。

解决方法：在 ossbrowser 工具界面，单击设置，修改上传下载设置。



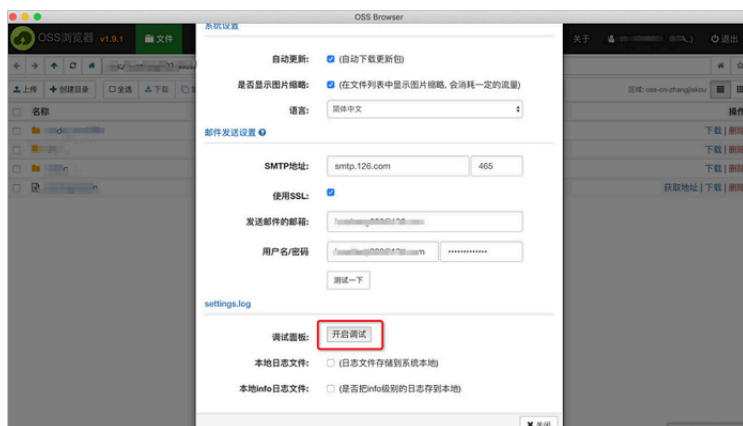
如何打开调试模式

您可以通过以下两种方式打开调试模式：

- 方式一：通过单击左上角应用 Logo 图标连续 10 次，弹出调试面板。



- 方式二：通过单击设置页面的开启调试，弹出调试面板。



更多常见问题

有关 ossbrowser 使用的更多常见问题，请参考[常见问题](#)。

3 命令行工具ossutil

3.1 概述

ossutil是以命令行方式管理OSS数据的工具，提供方便、简洁、丰富的Bucket和Object管理命令，支持Windows、Linux、Mac平台。

通过ossutil，您可以进行如下操作：

- 管理Bucket，如：创建、列举或删除Bucket等。
- 管理Object，如：上传、下载、列举、拷贝和删除Object等。
- 管理碎片（part），如：列举和删除part等。

安装ossutil

ossutil的下载及安装操作请参见[#unique_28](#)。

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
#unique_29	将本地文件内容以追加上传的方式上传到OSS中的appendable Object中。
#unique_30	添加、修改、查询、删除Bucket的加密配置。
#unique_31	添加、修改、查询、删除存储空间（Bucket）的Bucket policy配置。
#unique_32	添加、修改、查询、删除Bucket的标签配置。
#unique_33	将文件内容输出到ossutil。
#unique_34	创建配置文件来存储OSS访问信息。
#unique_35	添加、修改、查询、删除Bucket的CORS配置。
#unique_36	用于测试Bucket是否允许指定的跨域访问请求。
#unique_37	用于上传、下载、拷贝文件。
#unique_38	创建符号链接（软链接）。
#unique_39	用于获取指定存储空间（Bucket）或者指定文件（Object）/文件目录所占的存储空间大小。
#unique_40	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。

名称	描述
#unique_41	用于计算本地文件的CRC64或MD5。
help	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。
#unique_43	命令用于添加、修改、查询、删除生命周期规则配置。
#unique_44	列出没有完成分片上传的Object的分片信息。
#unique_45	添加、修改、查询、删除Bucket的日志管理配置。
#unique_46	列举Bucket、Object和Part。
#unique_47	创建Bucket。
#unique_48	在Bucket内创建文件目录。
#unique_49	添加、修改、查询或删除对象（Object）的标签配置。
#unique_50	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。
#unique_51	读取符号链接（软链接）文件的描述信息。
#unique_52	添加、修改、查询、删除Bucket的防盗链配置。
#unique_53	恢复冷冻状态的Object为可读状态。
#unique_54	设置或查询存储空间（Bucket）的请求者付费模式配置。
#unique_55	删除Bucket、Object和Part。
#unique_56	设置Bucket或Object的访问权限（ACL）。
#unique_57	设置已上传Object的元信息。
#unique_58	用于生成经过签名的url供第三方用户访问Bucket内的Object。
#unique_59	获取指定Bucket或Object的描述信息。
#unique_60	用于更新ossutil版本。
#unique_61	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。

3.2 下载和安装

本文主要介绍如何下载和安装ossutil。

版本和运行环境

- 当前版本：1.6.7
- 软件源码：[ossutil](#)

- 运行环境
 - Windows/Linux/Mac
 - 支持架构：x86（32bit、64bit）

下载地址

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



注意：

复制链接的时候，会自动加上spm编码内容，使用wget命令下载的时候，注意删除链接中？
*spm=xxxx*内容。

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

快速安装

根据您的操作系统选择对应版本的软件下载并配置ossutil。

- Linux系统（以64位系统为例）

1. 下载工具：

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

2. 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：

```
./ossutil64 config
```

该命令将创建一个配置文件，在其中存储配置信息。 请输入配置文件路径（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: http://oss-cn-shenzhen.aliyuncs.com

请输入accessKeyID: yourAccessKeyID

请输入accessKeySecret: yourAccessKeySecret

请输入stsToken:

- endpoint: 填写Bucket所在地域的域名信息, 可参考[访问域名和数据中心](#)。
- accessKeyID: 查看方式请参考[#unique_63](#)。
- accessKeySecret: 查看方式请参考[#unique_63](#)。
- stsToken: 非必配项, 若采用STS临时授权方式访问OSS需要配置该项, 否则置空即可。stsToken生成方式参考[临时访问凭证](#)。



说明:

更详细的配置文件说明可参考[#unique_65](#)。

- Windows系统（以64位系统为例）：

1. 下载工具。
2. 将工具解压到指定的文件夹, 并双击运行ossutil.bat文件。
3. 生成配置文件, 配置参数参考Linux系统的配置参数:

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

- Mac系统（以64位系统为例）：

1. 下载工具。

```
curl -o ossutilmac64 http://gosspublic.alicdn.com/ossutil/1.6.7/ossutilmac64
```

2. 修改文件执行权限:

```
chmod 755 ossutilmac64
```

3. 生成配置文件, 配置参数参考Linux系统的配置参数:

```
./ossutilmac64 config
```

3.3 常用命令

3.3.1 appendfromfile

appendfromfile命令用于将本地文件内容以追加上传的方式上传到OSS中的appendable Object中。



说明:

追加上传详情请参见[#unique_68](#)。

命令格式

```
./ossutil appendfromfile local_file_name oss://bucket/object [--meta=meta-value] [--payer requester]
```

如果Object不存在，可通过附加--meta选项设置Object的meta信息，例如--meta "X-Oss-Meta-Author:test"。如果Object已经存在，不可以在追加上传时附加--meta选项。



注意：

- 通过追加上传方式上传的Object，必须手动指定Object的名称。
- 只有通过追加上传创建的Object才可以后续继续被追加上传。
- 您可以通过[#unique_57](#)命令修改Object的meta信息。

使用示例

通过追加上传上传文件：

```
./ossutil appendfromfile /file/test.txt oss://bucket1/test.txt
total append 64(100.00%) byte,speed is 0.00(KB/s)
local file size is 64,the object new size is 64,average speed is 0.34(KB/s)
```

常用选项

您可以在使用appendfromfile命令时附加如下选项：

选项名称	描述
--meta	设置Object的meta信息。如果Object已经存在，不可附加此项。
--maxupspeed	设置最大上传速度，单位：KB/s，缺省值为0（不限速）。
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.2 bucket-encryption

bucket-encryption命令用于添加、修改、查询、删除Bucket的加密配置。



说明:

Bucket加密介绍请参见[#unique_71](#)。

命令格式

- 添加/修改Bucket加密配置

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

- --sse-algorithm: 选择Bucket的加密方式，可选值为KMS和AES256。
- --kms-masterkey-id: 设置指定的CMK ID进行加密。当--sse-algorithm值为AES256时，不可附加此项；值为KMS时，根据您的需求选择是否添加。



说明:

使用指定CMK ID加密目前处于公测中，请联系[售后技术支持](#)开通试用。

若Bucket未设置加密，此命令将为Bucket添加指定的加密方式；若Bucket已配置加密，此命令将更改Bucket加密配置。

- 获取Bucket加密配置

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

- 删除Bucket加密配置

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

使用示例

- 添加/修改Bucket加密配置

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

- 获取Bucket加密配置

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

- 删除Bucket加密配置

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

常用选项

您可以在使用bucket-encryption命令时附加如下选项：

选项名称	描述
--sse-algorithm	表示服务端加密算法，取值为KMS或AES256。
--kms-masterkey-id	设置指定的CMK ID进行加密。当--sse-algorithm值为AES256时，不可附加此项；值为KMS时，根据您的需求选择是否添加。
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：添加或修改配置。· get：获取配置。· delete：删除配置。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.3 bucket-policy

bucket-policy命令用于添加、修改、查询、删除存储空间（Bucket）的Bucket policy配置。



说明:

Bucket policy的功能介绍请参见[#unique_73](#)。

命令格式

- 添加/修改Bucket Policy配置

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

ossutil从配置文件`local_json_file`中读取Bucket Policy配置，并在Bucket中添加对应规则。若Bucket已有Bucket Policy配置，新的配置将覆盖原有配置。



说明:

`local_json_file`是一个Json格式的文件，配置举例如下：

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

- 获取Bucket Policy配置

```
./ossutil bucket-policy --method get oss://bucket [local_json_file] [options]
```

`local_json_file`为文件路径参数。若填写，则将Bucket Policy的配置保存为本地文件；若置空，则将Bucket Policy的配置输出到屏幕上。

- 删除Bucket Policy配置

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

使用示例

- 添加Bucket Policy配置, 允许指定IP的匿名用户访问Bucket内所有资源

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

*policy.json*文件内容为:

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

- 获取Bucket Policy配置

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

```
}  
]  
}  
}  
}  
]  
}
```

- 删除Bucket Policy配置

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

常用选项

您可以在使用bucket-policy命令时，附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：添加或修改配置。· get：获取配置。· delete：删除配置。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如https://120.79.***.***:3128、socks5://120.79.***.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.4 bucket-tagging

bucket-tagging命令用于添加、修改、查询、删除Bucket的标签配置。



说明：

Bucket标签介绍请参见[#unique_75](#)。

命令格式

- 添加/修改Bucket标签

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

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

- 只有Bucket的拥有者及授权子账户才能为Bucket设置用户标签，否则返回403 Forbidden 错误，错误码：AccessDenied。
 - 最多可设置20对Bucket用户标签（Key-Value对）。
 - Key最大长度为64字节，不能以http://、https://、Aliyun为前缀，且不能为空。
 - Value最大长度为128字节，可以为空。
 - Key和Value必须为UTF-8编码。
- 查询Bucket标签

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

- 删除Bucket标签

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

使用示例

- 添加Bucket标签

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

- 查询Bucket标签

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

- 删除Bucket标签

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

常用选项

您可以在使用bucket-tagging命令时附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：添加或修改配置。· get：获取配置。· delete：删除配置。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***.3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.5 cat

cat命令可以将文件内容输出到ossutil。

命令格式

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



说明：

仅建议使用此命令查看文本文件内容。

使用示例

查看指定文件内容：

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

0.273027(s) elapsed
```

常用选项

您可以在使用cat命令时附加如下选项：

选项名称	描述
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.**.**:3128、socks5://120.79.**.**:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.6 config

config命令用于创建配置文件来存储OSS访问信息。您可以在使用其他命令时添加-c选项，在访问OSS时提供访问信息。

命令格式

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



说明:

该命令有两种用法，交互式和非交互式。推荐用法为交互式，因为交互式用法拥有更好的安全性。

使用示例

· 交互式配置

```
./ossutil config
```

该命令将创建一个配置文件，在其中存储配置信息。请输入配置文件路径（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: http://oss-cn-shenzhen.aliyuncs.com

请输入accessKeyID: yourAccessKeyID

请输入accessKeySecret: yourAccessKeySecret

请输入stsToken:

- endpoint: 填写Bucket所在地域的域名信息, 可参考[访问域名和数据中心](#)。
 - accessKeyID: 查看方式请参考[#unique_63](#)。
 - accessKeySecret: 查看方式请参考[#unique_63](#)。
 - stsToken: 非必配项, 若采用STS临时授权方式访问OSS需要配置该项, 否则置空即可。stsToken生成方式参考[临时访问凭证](#)。
- 非交互式配置

```
./ossutil config -e oss-cn-beijing.aliyuncs.com -i LTAIbZcdVCmQ****  
-k D26oqKBudxDRBg8Wuh2EWDBrM0**** -L CH -c /myconfig
```

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

配置文件格式

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_key_id  
    accessKeySecret = your_key_secret  
    stsToken = your_sts_token  
    outputDir = your_output_dir  
[Bucket-Endpoint]  
    bucket1 = endpoint1  
    bucket2 = endpoint2  
    ...  
[Bucket-Cname]  
    bucket1 = cname1  
    bucket2 = cname2  
    ...  
[AkService]  
    ecsAk=http://100.100.100.200/latest/meta-data/Ram/security-  
credentials/EcsRamRoleTesting
```

- Bucket-Endpoint: 对每个指定的Bucket单独配置Endpoint, 当对某Bucket进行操作时, ossutil会在该选项中寻找该Bucket对应的Endpoint, 如果找到, 该Endpoint会覆盖Credentials选项中的endpoint。
- Bucket-Cname: Bucket-Cname为每个指定的Bucket单独配置CNAME域名(CDN加速域名), 此配置会优先于配置文件中Bucket-Endpoint及Credentials选项中endpoint的配置。关于CNAME域名的更多信息请参见[配置CNAME](#)。
- AkService: 此项默认不增加, 如果您希望使用ECS实例绑定的RAM角色操作OSS的话, 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，accessKeyID、accessKeySecret、stsToken可不配置；如果配置了accessKeyID，那么AkService的配置将不会生效，以配置的accessKeyID、accessKeySecret、stsToken进行身份校验。ECS实例绑定RAM角色请参见通过控制台使用实例RAM角色。ECS实例绑定RAM角色请参见[#unique_79](#)。

**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，但配置文件中该项配置仍然起效，您可以在配置文件中对每个Bucket单独指定Endpoint或CNAME。
- ossutil工具可以在多个地方指定Endpoint，生效优先级为：`--endpoint`（命令选项） > Bucket-Cname > Bucket-Endpoint > endpoint（Credentials项）。您在输入操作命令时，如果指定了`--endpoint`（可简写为`-e`）选项，以`--endpoint`的值为准。若未指定，则会在配置文件中依次查找Bucket-Cname、Bucket-Endpoint、Credentials内的配置，以高优先级的Endpoint配置为准。

常用选项

您可以在使用config命令时，指定如下选项来生成指定的配置项：

选项名称	描述
<code>-e, --endpoint</code>	设置配置文件中Credentials选项的endpoint项。
<code>-i, --access-key-id</code>	设置配置文件中Credentials选项的accessKeyID项。
<code>-k, --access-key-secret</code>	设置配置文件中Credentials选项的accessKeySecret项。
<code>-t, --sts-token</code>	设置配置文件中Credentials选项的stsToken项，非必填项。
<code>--output-dir</code>	指定输出文件所在的目录。输出文件目前包含： <code>cp</code> 命令批量拷贝文件出错时所产生的report文件。默认值为：当前目录下的 <code>ossutil_output</code> 目录。
<code>-L, --language</code>	设置ossutil工具的语言。默认值：CH，取值范围：CH、EN。若设置成CH，请确保您的系统编码为UTF-8。
<code>--loglevel</code>	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。

**说明：**

更多通用选项请参见[#unique_69](#)。

3.3.7 cors

`cors`命令用于添加、修改、查询、删除Bucket的跨域资源共享（CORS）配置。



说明:

CORS介绍请参见[#unique_81](#)。

命令格式

- 添加/修改CORS配置

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

若Bucket未配置CORS，ossutil从配置文件`local_xml_file`中读取CORS配置，并在Bucket中添加对应规则；若Bucket已配置CORS，ossutil将Bucket的CORS配置修改为配置文件内的配置。



说明:

`local_xml_file`是一个xml格式的文件，举例如下：

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

- 获取CORS配置

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

`local_xml_file`为文件路径参数。若填写，则将CORS的配置保存为本地文件；若置空，则将CORS的配置输出到屏幕上。

- 删除CORS配置

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

使用示例

- 添加CORS配置

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

- 获取CORS配置

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

- 删除CORS配置

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

常用选项

您可以在使用cors命令时附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：添加或修改配置。· get：获取配置。· delete：删除配置。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.8 cors-options

cors-options命令用于测试存储空间（Bucket）是否允许指定的跨域访问请求。



说明：

- 跨域访问详情请参见[#unique_83](#)。
- 设置跨域访问的命令请参见[#unique_84](#)。

命令格式

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

cors-options命令向OSS发送http options请求，测试目标OSS是否允许自己的跨域请求：

- --acr-method：http请求头Access-Control-Request-Method的值，可选值为：GET、PUT、POST、DELETE、HEAD。
- --origin：http请求头Origin字段的值，表示请求来源域，用来标示跨域请求。
- --acr-headers：http请求头Access-Control-Request-Headers的值，表示在实际请求中会用到的除简单头部之外的头。如果有多个取值，各个header用英文的逗号（,）分隔，再加上双引号。例如：--acr-headers "header1,header2,header3"。

使用示例

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

常用选项

您可以在使用cors-options命令时附加如下选项：

选项名称	描述
--origin	http请求头Origin字段的值，表示请求来源域，用来标示跨域请求。
--acr-method	http请求头Access-Control-Request-Method的值，可选值为：GET、PUT、POST、DELETE、HEAD。
--acr-headers	http请求头Access-Control-Request-Headers的值，表示在实际请求中会用到的除简单头部之外的头。如果有多个取值，各个header用英文的逗号（,）分隔，再加上双引号。例如：--acr-headers "header1,header2,header3"。
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。

选项名称	描述
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***.3128、socks5://120.79.***:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.9 cp

cp命令用于上传、下载、拷贝文件。

命令格式

- 上传文件

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

- 下载文件

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

- 拷贝文件

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

使用示例

- 上传文件

- 上传单个文件

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

- 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...，如下将当前目录下文件a.txt上传并设置其meta信息：

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



说明：

更多关于meta设置的信息请参见[#unique_57](#)。

- 上传文件夹

使用cp命令时增加-r选项可以将目标文件夹上传到OSS。命令如下：

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



说明：

若上传目标对象为符号链接（软链接），且指向本地文件夹，则使用cp命令上传时，应当给软链接加上正斜线（/）。

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

- 上传文件/文件夹并设置限速

上传文件时，指定--maxupspeed选项，可设置上传时的最高速度，单位为KB/s，缺省为0（不限速）。

■ 上传文件并设置限速为1MByte/s

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

■ 上传文件夹并设置限速为1MByte/s

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

- 批量上传符合条件的文件

您可以使用--include/--exclude，在cp操作时批量选择对应条件的文件。

--include/--exclude选项支持格式：

■ *：通配符，匹配所有字符。例如：*.txt表示匹配所有txt格式的文件。

■ ?：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]：匹配不在序列的任意字符，例如：abc[!0-7].jpg，表示匹配文件名不为abc0.jpg~abc7.jpg的文件。



说明：

■ 不支持带目录的格式，例如：--include "/usr/test/.jpg"。

■ 指定--include/--exclude选项时，需同时指定--recursive (-r) 选项。

■ 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。

示例：当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`为txt文件，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`为txt文件，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。

示例：

■ 上传所有文件格式为`txt`的文件：

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

■ 上传所有文件名包含`abc`且不是`jpg`和`txt`格式的文件：

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

- 上传文件并指定存储类型

您可以在上传文件时，通过`--meta`选项修改对象存储类型。

■ 上传单个文件并指定存储类型为IA（低频访问）类型

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

■ 上传文件目录并指定存储类型为IA类型：

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

- 上传文件并指定服务器端加密方式

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，关于服务器端加密功能介绍请参见[#unique_86](#)。

■ 上传文件并指定加密方式为AES256

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

■ 上传文件并指定加密方式为KMS

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

使用KMS加密时，OSS会为这个文件在KMS平台上创建一个主密钥，会产生少量KMS密钥API调用费用，详情请参见[KMS计费标准](#)。

- 上传文件夹并跳过已有文件

批量上传时，若指定--update（可缩写为-u）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才会执行上传操作。命令如下：

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

该选项可用于当批量上传失败重传时，跳过已经成功的文件，实现增量上传。

- 上传文件夹并生成快照信息

批量上传时，若指定--snapshot-path选项，ossutil在指定的目录下生成文件上传的快照信息，在下一次指定该选项上传时，ossutil会读取指定路径下的快照信息进行增量上传。命令如下：

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



注意：

- --snapshot-path选项用于在某些场景下加速增量上传/下载批量文件（拷贝不支持该选项）。例如，文件数较多且两次上传期间没有其他用户更改OSS上对应的Object。
- --snapshot-path选项通过在本地记录成功上传/下载的文件的本
地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--update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--update选项来增量上传/下载。

■ `--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。

- 上传文件到开通了请求者付费模式的Bucket

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

- 下载文件

- 下载单个文件

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

- 指定范围下载文件

下载文件时，可以通过`--range`选项进行指定范围下载。

示例：将`test1.txt`的第10到第20个字符作为一个文件载到本地。

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

```
0.290769(s) elapsed
```

- 从开通了请求者付费模式的 Bucket 下载文件

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

- 下载文件夹

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

- 下载文件夹并指定--update选项

批量下载时，若指定--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才会执行下载操作。命令如下：

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

该选项可用于当批量下载失败重传时，跳过已经下载成功的文件，实现增量下载。

- 下载文件夹并生成快照信息

批量下载时，若指定--snapshot-path选项，ossutil在指定的目录下生成文件下载的快照信息，在下一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[--snapshot-path选项](#)。命令如下：

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

- 批量下载符合指定条件的文件

您可以使用--include/--exclude参数，在下载时选定符合条件的文件。详情请参见[过滤选项](#)。

■ 下载所有文件格式不为jpg的文件

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

■ 下载所有文件名包含abc且不是jpg和txt格式的文件

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

- 拷贝文件

目前只支持拷贝文件，不支持拷贝未完成的Multipart，不支持跨region拷贝文件。

- 拷贝单个文件

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

- 拷贝单个文件并重命名

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

- 批量拷贝符合指定条件的文件

您可以使用`--include/--exclude`，在拷贝时选定符合条件的文件。详情请参见[过滤选项](#)。

■ 拷贝所有文件格式不为jpg的文件

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

■ 拷贝所有文件名包含abc且不是jpg和txt格式的文件

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

- 拷贝文件并修改文件存储类型

■ 修改已有文件的存储类型为Archive（归档存储）类型

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

■ 修改已有文件目录下所有文件的文件类型为Standard（标准存储）类型

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



注意:

- 存储类型为归档类型的文件不能通过cp命令直接转换成其他类型，必须先通过[#unique_53](#)命令将该文件解冻，之后再使用cp命令转换文件类型。

■ 使用cp覆写文件时涉及到数据覆盖操作。如果低频访问型或归档型文件分别在创建后30和60天内被覆盖，则它们会产生提前删除费用。详情请参见[计量项和计费项](#)。

- 拷贝文件并指定服务器端加密方式

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请参见[#unique_86](#)。

■ 拷贝文件并指定加密方式为AES256

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

■ 拷贝文件并指定加密方式为KMS

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

使用KMS加密时，OSS会为这个文件在KMS平台上创建一个主密钥，会产生少量KMS密钥API调用费用，详情请参见[KMS计费标准](#)。

- 拷贝单个文件并指定--meta选项

拷贝文件的同时可以使用 --meta 选项设置 Object 的 meta 信息，其内容格式为 header: value#header:value...。

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

- 同region不同Bucket之间的文件拷贝

拷贝文件时，搭配 -r 选项可以实现批量文件拷贝功能。

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

- 同region不同Bucket之间的增量文件拷贝

批量拷贝时，若指定--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：

```
./ossutil cp oss://your_src_bucket/path1/ oss://your_dest_bucket/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。

- 从开通了请求者付费模式的Bucket拷贝文件到普通Bucket

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

- 从普通Bucket拷贝数据到开通了请求者付费模式的Bucket

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

- 上传/下载/拷贝文件的性能调优

在cp命令中，通过--jobs项和--parallel项控制并发数。当ossutil自行设置的默认并发达不到用户的性能需求时，您可以自行调整该两个选项来升降性能。

- 使用--jobs项来控制多个文件上传/下载/拷贝时，文件间启动的并发数。
- 使用--parallel项来控制分片上传/下载/拷贝一个大文件时，每一个大文件启动的并发数。

默认情况下，ossutil会根据文件大小来计算parallel个数（该选项对于小文件不起作用，进行分片上传/下载/拷贝的大文件文件阈值可由--bigfile-threshold选项来控制）。当进行批量大文件的上传/下载/拷贝时，实际的并发数为jobs个数乘以parallel个数。



警告：


- 通常情况下，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。
- 如果并发数调得太大，由于线程间资源切换及抢夺等，ossutil上传/下载/拷贝性能可能会下降。并发数过大可能会产生EOF错误。所以请根据实际的机器情况调整--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。

常用选项

您可以在使用cp命令时附加如下选项：

选项名称	描述
-r, --recursive	递归进行操作。当指定该选项时，命令会对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/下载/拷贝操作。

选项名称	描述
--output-dir	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。默认值为：当前目录下的ossutil_output目录。
--bigfile-threshold	开启大文件断点续传的文件大小阈值，单位为Byte，默认值：100MByte，取值范围：0-9223372036854775807（Byte）。
--part-size	分片大小，单位为Byte。默认情况下ossutil根据文件大小自行计算合适的分片大小值。如果有特殊需求或者需要性能调优，可以设置该值，取值范围：1-9223372036854775807（Byte）。
--checkpoint-dir	checkpoint目录的路径（默认值为：.ossutil_checkpoint），断点续传操作失败时，ossutil会自动创建该目录，并在该目录下记录checkpoint信息，操作成功会删除该目录。如果指定了该选项，请确保所指定的目录可以被删除。
--range	下载文件时，指定文件下载的范围，格式为：3-9或3-或-9。
--encoding-type	输入或者输出的文件名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示文件名未经过编码。Bucket名不支持url编码。
--include	包含对象匹配模式，如：*.jpg。
--exclude	不包含对象匹配模式，如：*.txt。
--meta	设置Object的meta为[header:value#header:value...]，如：Cache-Control:no-cache#Content-Encoding:gzip。更多详情请参见 #unique_57 。
--acl	设置Object的访问权限，默认为default。可配置项为： <ul style="list-style-type: none">· default：继承Bucket· private：私有· public-read：公共读· public-read-write：公共读写

选项名称	描述
--snapshot-path	<p>批量上传/下载时，若指定--snapshot-path选项，ossutil在指定的目录下生成文件上传/下载的快照信息，在下次指定该选项上传/下载时，ossutil会读取指定路径下的快照信息进行增量上传/下载。</p> <div>  注意： <ul style="list-style-type: none"> · --snapshot-path选项用于在某些场景下加速增量上传/下载批量文件（拷贝不支持该选项）。例如，文件数较多且两次上传期间没有其他用户更改OSS上对应的Object。 · --snapshot-path命令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--update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--update选项来增量上传/下载。 · --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。 </div>
--disable-crc64	该选项关闭CRC64，默认情况下，ossutil进行数据传输都打开CRC64校验。
--maxupspeed	最大上传速度，单位：KB/s，缺省值为0（不受限制）。
--payer	请求的支付方式，如果为请求者付费模式，可以将该值设置成requester。
--partition-download	分区下载使用。一个ossutil命令下载一个分区，其值格式为“分区编号：总分区数”，例如1：5，表示当前ossutil下载分区1，总共有5个分区，分区编号从1开始，Object的分区规则由工具内部算法决定。利用该选项，将待下载的Object分成多个区，可以由多个ossutil命令同时下载，每个ossutil命令下载各自的分区，多个ossutil命令可以并行在不同机器上执行。
-j, --jobs	多文件操作时的并发任务数，默认值：3，取值范围：1-10000。

选项名称	描述
--parallel	单文件内部操作的并发任务数，取值范围：1-10000，默认将由ossutil根据操作类型和文件大小自行决定。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.10 create-symlink

create-symlink命令用于创建符号链接（软链接）。



说明：

关于软链接的介绍请参见[#unique_90](#)。

命令格式

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

使用示例

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

此命令为bucket1内path目录下的a.txt创建一个名为b，存放在bucket1根目录下的软链接文件。



说明：

创建符号链接时：

- 不检查目标文件是否存在。
- 不检查目标文件类型是否合法。
- 不检查目标文件是否有权限访问。

- 以上检查在GetObject等需要访问目标文件的API接口时进行。
- 如果试图添加的文件已经存在，并且有访问权限，新添加的文件将覆盖原来的文件。

可通过[#unique_59](#)或[#unique_51](#)命令可以查看软链接的目标文件。

常用选项

您可以在使用create-symlink命令时附加如下选项：

选项名称	描述
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.**.*:3128、socks5://120.79.**.*:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.11 du

du命令用于获取指定存储空间（Bucket）或者指定文件（Object）/文件目录所占的存储空间大小。

命令格式

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

使用示例

- 查询指定Bucket所占存储空间大小

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

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

- 查看指定文件目的所占存储空间大小

```
./ossutil du oss://bucket1/test/  
object count:2  object sum size:64482  
part count:0    part sum size:0  
total du size(byte):64482  
  
0.176018(s) elapsed
```

- 查看指定Object所占存储空间大小

```
./ossutil du oss://bucket1/test.txt  
object count:1  object sum size:27856  
part count:0    part sum size:0  
total du size(byte):27856  
  
0.241024(s) elapsed
```

- 查看指定前缀的所有Object所占存储空间大小

```
./ossutil du oss://bucket1/test  
object count:3  object sum size:92338  
part count:0    part sum size:0  
total du size(byte):92338  
  
0.184018(s) elapsed
```

常用选项

您可以在使用du命令时，附加如下选项：

选项名称	描述
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***.3128、socks5://120.79.***:1080。
--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。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.12 getallpartsize

getallpartsize命令用于获取存储空间（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。

命令格式

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

使用示例

列举所有未完成上传的Multipart Object的分片信息以及总大小：

```
./ossutil getallpartsize oss://bucket1
PartNumber      UploadId          Size(Byte)
Path
1               F18A92392DFD4B3FA897C267829FE417    52428800
oss://bucket1/test/test1.txt
2               F18A92392DFD4B3FA897C267829FE417    52428800
oss://bucket1/test/test1.txt
3               F18A92392DFD4B3FA897C267829FE417    52428800
oss://bucket1/test/test1.txt
4               F18A92392DFD4B3FA897C267829FE417    52428800
oss://bucket1/test/test1.txt
5               F18A92392DFD4B3FA897C267829FE417    52428800
oss://bucket1/test/test1.txt
6               F18A92392DFD4B3FA897C267829FE417    52428800
oss://bucket1/test/test1.txt

total part count:6      total part size(MB):300.00
```

常用选项

您可以在使用getallpartsize命令时附加如下选项：

选项名称	描述
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***.3128、socks5://120.79.***.1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.13 hash

hash命令用于计算本地文件的CRC64或MD5。

命令格式

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

--type选项用来控制计算的类型，可选值为crc64和md5，默认为crc64。



说明:

- OSS文件的CRC64和Content-MD5值可通过[#unique_59](#)命令查看到，参考X-Oss-Hash-Crc64ecma字段和Content-Md5字段。
- 若文件在OSS支持CRC64功能之前上传，则无法使用stat命令查看CRC64值。
- 对于追加上传和分片上传类型的文件，无法使用stat命令查看Content-MD5值。
- 计算类型为md5时，会同时输出文件的MD5以及Content-MD5值。Content-MD5值其实是先计算5值获得128比特位数字，然后对该数字进行base64编码得到的值。
- CRC64的计算标准参考[ECMA-182标准](#)。
- 关于Content-MD5的更多信息，请参考[RFC1864](#)。

使用示例

- 计算本地文件的CRC64

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

- 计算本地文件的md5

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

常用选项

您可以在使用hash命令时附加如下选项：

选项名称	描述
--type	计算的类型，默认值：crc64，取值范围：crc64、md5。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.14 help

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。

命令格式

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

使用示例

- 获取所有命令帮助信息

```
./ossutil help
```

- 获取cp命令帮助信息

```
./ossutil help cp
```

- 查看所有选项信息

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

- 切换为中文显示ls命令的帮助

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

3.3.15 lifecycle

lifecycle命令用于添加、修改、查询、删除生命周期规则配置。



说明:

生命周期规则介绍请参见[#unique_95](#)。

命令格式

- 添加/修改生命周期规则配置

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

ossutil从配置文件`local_xml_file`中读取生命周期规则配置，并在Bucket中添加对应规则。若添加的生命周期规则ID已经存在，则会覆盖已有ID的生命周期规则配置。



说明:

配置文件是一个xml格式的文件，可以配置文件过期后转为低频访问、归档存储，或直接删除文件。您可以选择部分规则进行配置，完整配置举例如下：

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

配置文件内容介绍请参见[#unique_96](#)。

- 获取生命周期规则配置

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

*local_file*为文件路径参数。若填写，则将生命周期规则配置保存为本地文件；若置空，则将生命周期规则配置输出到屏幕上。

- 删除生命周期规则配置

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

使用示例

- 添加生命周期规则，将已上传10天，且前缀为*test/*的文件转为低频访问类型

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

*lifecycle.xml*文件内容为：

```
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>1</ID>
    <Prefix>test/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>10</Days>
      <StorageClass>IA</StorageClass>
    </Transition>
  </Rule>
```

```
</LifecycleConfiguration>
```

- 添加生命周期规则，将文件最后修改日期2019年1月1日前，且前缀为test/的文件删除

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

lifecycle.xml文件内容为：

```
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>2</ID>
    <Prefix>test/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <CreatedBeforeDate>2019-01-01T00:00:00.000Z</CreatedBeforeDate>
    >
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

- 获取生命周期规则配置

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

- 删除生命周期规则配置

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

常用选项

您可以在使用lifecycle命令时附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：添加或修改配置。· get：获取配置。· delete：删除配置。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.**.*:3128、socks5://120.79.**.*:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.16 listpart

listpart命令用于列出没有完成分片上传的Object的分片信息。

命令格式

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

您需要先通过[ls -m](#)命令查看Bucket的Object和UploadID信息，再用本命令查看分片的详细信息。

使用示例

列举单个未完成分片上传的Object的分片信息：

```
./ossutil listpart oss://bucket/object 15754AF7980C4DFB8193F190837520  
BB
```

常用选项

您可以在使用listpart命令时附加如下选项：

选项名称	描述
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.17 logging

logging命令用于添加、修改、查询、删除Bucket的日志管理配置。



说明:

日志管理详情请参见[#unique_99](#)。

命令格式

- 添加/修改日志管理配置

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

若Bucket未开启日志管理，此命令将Bucket的访问日志以Object的形式保存到target-bucket中；若Bucket已开启日志管理，此命令可修改日志记录的存储位置。

prefix参数可指定日志记录存储的目录和前缀。若填写，日志将被记录到target-bucket的指定目录下；若不填写，则保存到target-bucket的根目录下。日志文件命名规则请参见[日志记录命名规则](#)。

- 查看日志管理配置

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

local_xml_file为文件路径参数。若填写，则将日志管理的配置保存为本地文件；若置空，则将日志管理的配置输出到屏幕上。

- 删除日志管理配置

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

使用示例

- 添加日志管理配置

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

- 查看日志管理配置

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

- 删除日志配置

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

常用选项

您可以在使用logging命令时附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：添加或修改配置。· get：获取配置。· delete：删除配置。

选项名称	描述
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none"> · info：输出提示信息日志。 · debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.**.*:3128、socks5://120.79.**.*:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.18 ls

ls命令用于列举存储空间（Bucket）、对象（Object）和碎片（Part）。

命令格式

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

使用示例

- 列举Bucket

```
./ossutil ls
CreationTime                               Region    StorageClass
BucketName
2016-10-21 16:18:37 +0800 CST               oss-cn-hangzhou    Archive
oss://bucket1
2016-12-01 15:06:21 +0800 CST               oss-cn-hangzhou    Standard
oss://bucket2
2016-07-20 10:36:24 +0800 CST               oss-cn-hangzhou    IA
oss://bucket3
Bucket Number is:3
0.252174(s) elapsed
```



说明:

您也可以使用./ossutil ls oss://命令列举Bucket。

- 分页列举Bucket

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

当Bucket数目太多时，可以使用--limited-num与--marker选项来分页列举Bucket。

- --limited-num用于控制分页展示条数。
- --marker用于控制分页从哪个Bucket开始列举，ossutil显示的结果从marker设定值之后按字母排序的第一个Bucket开始返回。该值一般为上一页查询显示的最后一个BucketName。

示例：分页列举前两个Bucket。

```
./ossutil ls oss:// --limited-num=1 -s
oss://bucket1
Bucket Number is:1
0.303869(s) elapsed
./ossutil ls oss:// --limited-num=1 -s --marker=bucket1
oss://bucket2
Bucket Number is:1
0.257636(s) elapsed
```

- 列举指定Bucket下所有的Object

```
./ossutil ls oss://bucket1
LastModifiedTime                               Size(B)  StorageClass  ETAG
Object Name
2016-12-0115:06:37 +0800 CST                    10363812      Standard
61DE142E5AFF9A6748707D4A77BFBCFB              oss://bucket1/a1
2016-12-0115:06:42 +0800 CST                    10363812      Standard
61DE142E5AFF9A6748707D4A77BFBCFB              oss://bucket1/a2
2016-12-0115:06:45 +0800 CST                    10363812      Standard
61DE142E5AFF9A6748707D4A77BFBCFB              oss://bucket1/a3
Object Number is:3
0.007379(s) elapsed
```

- 分页列举所有的Object

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

与分页列举Bucket类似，可以使用--limited-num与--marker选项来分页列举Object。示例，列举bucket1根目录下前两个Object：

```
./ossutil ls oss://bucket1 --limited-num=1
LastModifiedTime                               Size(B)  StorageClass  ETAG
Object Name
2016-12-0115:06:37 +0800 CST                    10363812      Standard
61DE142E5AFF9A6748707D4A77BFBCFB              oss://bucket1/a1
Object Number is:1
0.007379(s) elapsed
$./ossutil ls oss://bucket1 --limited-num=1 --marker=a1
LastModifiedTime                               Size(B)  StorageClass  ETAG
Object Name
2016-12-0115:06:42 +0800 CST                    10363812      Standard
61DE142E5AFF9A6748707D4A77BFBCFB              oss://bucket1/a2
```

```
Object Number is:1
0.008392(s) elapsed
```

- 列举符合条件的Object

您可以使用--include/--exclude参数，列举符合条件的Object。--include/--exclude参数详情请参见[过滤选项](#)。

- 列举所有文件格式不为jpg的Object

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

- 列举所有文件名包含abc且不是jpg和txt格式的Object

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

- 列举UploadID

使用加-m选项的ls命令列举对应前缀的Object下的所有未完成的Multipart上传任务的ID。

```
./ossutil ls oss://bucket1/obj1 -m
InitiatedTime          UploadID
  ObjectName
2017-01-13 03:45:26 +0000 CST      15754AF7980C4DFB8193F190837520BB
  oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST      2A1F9B4A95E341BD9285CC42BB950EE0
  oss://bucket1/obj1
UploadId Number is: 2
0.070070(s) elapsed
```

- 列举所有的Object和未完成的Multipart事件

使用加-a选项的ls命令列举对应前缀的Object下所有已上传完成的文件和未完成的Multipart上传任务的ID。

```
./ossutil ls oss://bucket1 -a
LastModifiedTime          Size(B)  StorageClass  ETAG
  ObjectName
2015-06-05 14:06:29 +0000 CST      201933      Standard
  7E2F4A7F1AC9D2F0996E8332D5EA5B41  oss://bucket1/dir1/obj11
2015-06-05 14:36:21 +0000 CST      201933      Standard
  6185CA2E8EB8510A61B3A845EAFE4174  oss://bucket1/obj1
2016-04-08 14:50:47 +0000 CST      6476984     Standard
  4F16FDAE7AC404CEC8B727FCC67779D6  oss://bucket1/sample.txt
Object Number is:3
InitiatedTime          UploadID
  ObjectName
2017-01-13 03:45:26 +0000 CST      15754AF7980C4DFB8193F190837520BB
  oss://bucket1/obj1
2017-01-13 03:43:13 +0000 CST      2A1F9B4A95E341BD9285CC42BB950EE0
  oss://bucket1/obj1
2017-01-13 03:45:25 +0000 CST      3998971ACAF94AD9AC48EAC1988BE863
  oss://bucket1/obj2
2017-01-20 11:16:21 +0800 CST      A20157A7B2FEC4670626DAE0F4C0073C
  oss://bucket1/tobj
UploadId Number is:4
```



```
0.191289(s) elapsed
```

- 以精简模式显示列举结果

```
./ossutil ls oss://bucket1 -s
oss://bucket1/a1
oss://bucket1/a2
oss://bucket1/a3
Object Number is:3
0.007379(s) elapsed
```

- 模拟目录方式列举

使用-d选项可以显示当前目录下的文件和子目录，而非递归显示所有子目录下的所有Object。

示例：

```
./ossutil ls oss://bucket1 -s -d
oss://bucket1/obj1
oss://bucket1/sample.txt
oss://bucket1/dir1/
Object and Directory Number is:3
0.119884(s) elapsed
```

- 列举当前目录下指定前缀的Object或文件目录

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

- 请求者付费模式下列举

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

常用选项

您可以在使用ls命令时附加如下选项指定列举模式及列举的内容：

选项名称	描述
-s	以精简模式列举。
-d	返回当前目录下的文件和子目录，而非递归显示所有子目录下的所有Object。
-m	指定操作的对象为bucket中未完成的Multipart事件，而非默认情况下的object。
-a	指定操作的对象为bucket中的object和未完成的Multipart事件。
--limited-num	返回结果的最大个数。与--marker同用可以分页列举目标。
--marker	列举Buckets时的marker，或列举objects或Multipart Uploads时的key marker。

选项名称	描述
--upload-id-marker	列举Multipart Uploads时的uploadID marker。
--payer	请求的支付方式，如果为请求者付费模式，可以将该值设置成requester。
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--include	包含对象匹配模式，如：*.jpg。
--exclude	不包含对象匹配模式，如：*.txt。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.19 mb

mb命令用于创建存储空间（Bucket）。

命令格式

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

使用示例

- 创建Bucket

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



注意：

Bucket名称唯一，若创建的Bucket名称已存在，会出现Bucket已存在的报错信息。

- 创建Bucket时指定访问权限

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

- 创建Bucket时指定存储类型

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

- 在指定地域创建Bucket

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

- 使用指定的配置文件创建Bucket

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

常用选项

您可以在使用mb命令时，指定如下选项来指定Bucket属性：

选项名称	描述
--acl	用来指定存储空间访问权限，默认为私有（private）。取值范围： <ul style="list-style-type: none">· private：私有· public-read：公共读· public-read-write：公共读写 ACL详情请参见 #unique_104 。
--storage-class	用来指定Bucket的默认存储类型。默认存储类型为标准存储类型（Standard）。取值范围： <ul style="list-style-type: none">· Standard：标准存储· IA：低频访问· Archive：归档存储 存储类型详情请参见 #unique_20 。
-c, --config-file	使用指定的config文件来创建Bucket。未附加该参数则表示使用默认配置文件创建Bucket。config配置请参见 #unique_65 。
-e, --endpoint	用来指定Bucket的地域（Region）。未附加该参数则表示使用默认配置文件中的Endpoint来指定Bucket的地域。
-L, --language	设置ossutil工具的语言，默认值：CH，取值范围：CH/EN，若设置成“CH”，请确保您的系统编码为UTF-8。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。

选项名称	描述
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.20 mkdir

mkdir命令可在存储空间内创建文件目录。

命令格式

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

使用示例

- 创建一个文件目录

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



说明：

- 文件目录需以正斜线 (/) 结尾，若未添加正斜线 (/)，ossutil 会自动添加。
- 若目录已存在，会出现报错。

- 创建一个多级目录

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

创建多级目录时，只会创建最后一级目录。例如此命令中，若将dir2/删除，且dir1/中无文件，则dir1/会消失。

常用选项

您可以在使用mkdir命令时附加如下选项：

选项名称	描述
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***.3128、socks5://120.79.***:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.21 object-tagging

object-tagging命令用于添加、修改、查询或删除对象（Object）的标签配置。



说明:

对象标签功能介绍请参见[#unique_107](#)。

命令格式

- 添加/修改对象标签配置

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

若Object未设置标签，此命令将为Object添加指定的标签；若Object已配置标签，此命令将覆盖Object原有标签。



说明:

- 单个文件可设置最多10个标签，Key不可重复。
- 每个Key长度不超过128字节，每个Value长度不超过256字节。
- Key和Value区分大小写。
- 标签合法字符集包括大小写字母、数字、空格和以下符号：

+ = . _ : /

- 只有Bucket拥有者和授权用户拥有读写对象标签的权限，该权限受对象ACL控制。
- 跨区域复制时，对象标签也会复制到目的端。

- 查询对象标签配置

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

- 删除对象标签配置

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

使用示例

- 添加对象标签配置

```
./ossutil object-tagging --method put oss://bucket1/test.jpg a#1 b#2 c#3
0.168034(s) elapsed
```

- 查询对象标签配置

```
./ossutil object-tagging --method get oss://bucket1/test.jpg
object index    tag index      tag key        tag value      object
-----
1              0              "a"            "1"            123/test.jpg
1              1              "b"            "2"            123/test.jpg
1              2              "c"            "3"            123/test.jpg
0.228023(s) elapsed
```

- 删除对象标签配置

```
./ossutil object-tagging --method delete oss://bucket1/test.jpg
0.200020(s) elapsed
```

常用选项

您可以在使用object-tagging命令时附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none"> • put：添加或修改配置。 • get：获取配置。 • delete：删除配置。

选项名称	描述
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.**.**:3128、socks5://120.79.**.**:1080。
--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。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。
-j, --jobs	多文件操作时的并发任务数，默认值：3，取值范围：1-10000。
-r, --recursive	递归进行操作。当指定该选项时，命令会对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。
--encoding-type	输入或者输出的文件名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示文件名未经过编码。Bucket名不支持url编码。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.22 probe

probe命令是针对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。

命令格式

- 下载http_url地址到本地，并输出探测报告

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

通过文件URL将存储空间内的一个文件下载到本地来测试网络传输质量，并输出探测报告。

- --url：填写指定Bucket内文件的URL地址。

■ 公共读文件：直接输入文件URL，例如：`https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg`。

■ 私有文件：输入带签名的文件URL，并且在URL前后需加双引号（“”）。例如：“`https://bucketname.oss-cn-beijing.aliyuncs.com/myphoto.jpg`”

```
?Expires=1552015472&OSSAccessKeyId=TMP.xxxxxxxx5r9f1FV12y8_
Qis6LUVmvoSCUSs7aboCCHtydQ0axN32Sn-UvyY3AAAwLAIUarYNLcO87AKMEcE5O
3AxxxxxxoCFAQuRdZYyVFyq0W8QkGAN-bamUiQ&Signature=bIa4llbMbl
drl7rwckr%2FXXvTtxw%3D”。
```



说明:

如何文件URL请参考[获取URL](#)。

- --addr=domain_name (可选)：可在下载文件时同时向一个指定的域名或IP地址发起ping操作。不添加该选项，ossutil不进行额外的探测操作。
 - 增加--addr=选项，但是缺省值的话，默认向www.aliyun.com发起ping操作。
 - 增加--addr=选项，并指定一个域名或IP，则向指定的地址发起ping操作。
- file_name (可选)：为下载的文件指定存储路径。如果不输入file_name，则下载文件保存在当前目录下，文件名由工具自动判断；如果输入file_name，则file_name为文件名或者目录名，下载的文件名为file_name或者保存在file_name目录下。
- 下载指定Bucket中的Object，并输出探测报告

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

- --bucketname：填写Bucket名。
- --object= (可选)：填写指定的Object路径可下载指定的Object。例如：*path/myphoto.jpg*。不指定--object=，则工具会生成一个临时文件上传到bucket-name后再将其下载，下载结束后会将该临时文件从本地和Bucket中删除。
- 上传Object并输出探测报告

```
./ossutil probe --upload [file_name] --bucketname bucket-name [--
object=obj
```



```
ect_name] [--addr=domain_name] [--upmode]
```

- file_name (可选)：指定file_name可将指定的文件上传到bucket-name中；不指定file_name，则工具会生成一个临时文件上传到bucket-name中，探测结束后会将临时文件删除。
- --object= (可选)：输入文件名或文件路径，可指定Object上传后的名称，例如：*path/myphoto.jpg*。若不配置--object=，则由工具自动生成上传后的Object名称，探测结束后会将该临时Object删除。
- --upmode (可选)：指定上传的方式，缺省该项则使用正常上传。可选值为：
 - normal：正常上传
 - append：追加上传
 - multipart：分片上传

使用示例

- 下载http_url地址到本地，并输出探测报告

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

- 下载指定Bucket中的Object，并输出探测报告

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

- 上传探测并输出探测报告

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

查看探测报告

probe命令运行后，您可以看到任务执行的步骤及结果。

- 执行步骤后出现×表示没有通过，否则表示通过。
- 结果显示整个上传下载成功还是失败。当成功时，会给出文件的大小和上传下载时间；失败时，会给出导致错误的原因，或直接给出修改建议。



说明：

并不是每次错误的检测都能提示出修改建议，对于没有提示修改建议的检测，请根据错误码提示，并结合oss错误码[ErrorCode](#)进行问题排查。

probe命令执行完毕后，ossutil会在当前目录下生成一个以logOssProbe开头的文件，里面包含此次探测命令执行的详细信息。

常用选项

您可以在使用probe命令时附加如下选项：

选项名称	描述
--url	表示一个网络地址，ossutil会下载该地址。
--bucketname	OSS中Bucket的名称。
--object	OSS中Object的名称。
--addr	需要网络探测的域名，工具会对该域名进行ping等操作，默认值为www.aliyun.com。
--upmode	指定上传的方式，缺省该项则使用正常上传。可选值为： <ul style="list-style-type: none">· normal：正常上传· append：追加上传· multipart：分片上传
--upload	表示上传到OSS。
--download	表示从OSS下载。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.23 read-symlink

read-symlink命令用于读取符号链接（软链接）文件的描述信息。



说明：

- 关于软链接的介绍请参见[#unique_90](#)。

- 创建软链接的命令请参见[#unique_38](#)。

命令格式

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

此操作要求用户对该符号链接文件有读权限。返回的项中，X-Oss-Symlink-Target表示符号链接的目标文件。如果操作的Object并非符号链接文件，该操作返回错误：NotSymlink。

使用示例

查看object1的软链接信息：

```
./ossutil read-symlink oss://bucket1/object1
Etag           : 455E20DBFFF1D588B67D092C46B16DB6
Last-Modified  : 2017-04-17 14:49:42 +0800 CST
X-Oss-Symlink-Target : a
```

常用选项

您可以在使用create-symlink命令时附加如下选项：

选项名称	描述
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.24 referer

referer命令用于添加、修改、查询、删除Bucket的防盗链配置。



说明:

防盗链介绍请参见[#unique_112](#)。

命令格式

- 添加/修改防盗链配置

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

若Bucket未设置防盗链，此命令将添加防盗链配置；若Bucket已设置防盗链，此命令将修改防盗链配置。

- referer-value: 设置Referer白名单，仅允许指定的域名访问OSS资源。支持通配符星号 (*) 和问号 (?)，多个域名可用空格隔开。
- --disable-empty-referer: 选择是否允许Referer为空。增加此选项表示不允许Referer为空，反之表示允许Referer为空。如果不允许空Referer，则只有HTTP或HTTPS header中包含Referer字段的请求才能访问OSS资源。
- 查看防盗链配置

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

local_xml_file为文件路径参数。若填写，则将防盗链的配置保存为本地文件；若置空，则将防盗链的配置输出到屏幕上。

- 删除防盗链配置

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

使用示例

- 设置防盗链，且不允许Referer为空：

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

- 查看防盗链配置

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

- 删除防盗链配置

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

常用选项

您可以在使用referer命令时附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：添加或修改配置。· get：获取配置。· delete：删除配置。
--disable-empty-referer	选择是否允许Referer为空。增加此选项表示不允许Referer为空，反之表示允许Referer为空。如果不允许空Referer，则只有HTTP或HTTPS header中包含Referer字段的请求才能访问OSS资源。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.25 restore

restore命令用于恢复冷冻状态的对象（Object）为可读状态。

命令格式

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

使用示例

- 恢复单个冷冻状态的Object为可读状态

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

- 恢复指定前缀的所有状态的Object为可读状态

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



说明:

- restore操作解冻过程需要1分钟时间，解冻中无法读取文件。
- 解冻状态默认持续 1 天，对解冻状态的文件调用restore命令，会将文件的解冻状态延长1天，最多可以延长到7天，之后文件又回到初始时的冷冻状态。

常用选项

您可以在使用restore命令时附加如下选项：

选项名称	描述
-r, --recursive	递归进行操作。当指定该选项时，命令会对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。
-j, --jobs	多文件操作时的并发任务数，默认值：3，取值范围：1-10000。
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--output-dir=	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（关于report文件更多信息，请参考cp命令帮助）。默认值为：当前目录下的ossutil_output目录。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。

选项名称	描述
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.26 request-payment

request-payment设置或查询存储空间（Bucket）的请求者付费模式配置。



说明：

请求者付费模式的功能介绍请参见[#unique_115](#)。

命令格式

- 设置请求者付费模式

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

payment_parameter可选值为Requester和BucketOwner。

- Requester：表示开启请求者付费模式，访问此Bucket内资源产生的所有费用由请求者支付。
- BucketOwner：表示关闭请求者付费模式，访问此Bucket内资源产生的所有费用由Bucket拥有者支付。

- 查询请求者付费模式配置

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

使用示例

- 开启请求者付费模式

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

- 关闭请求者付费模式

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

- 查询请求者付费模式配置

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

常用选项

您可以在使用request-payment命令时附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：开启或关闭请求者付费模式。· get：获取配置。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***.3128、socks5://120.79.***.1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.27 rm

rm命令用于删除存储空间（Bucket）、对象（Object）或碎片（Part）。

命令格式

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

使用示例

- 删除空Bucket

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



注意:

- 删除Bucket必须设置-b选项。
- 被删除的Bucket可能被其他用户重新创建，您不再拥有该Bucket。

- 清除Bucket数据并删除Bucket

如果Bucket中有Object或Multipart等数据，需要先删除所有数据再删除Bucket。命令如下：

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



警告:

该命令将清除Bucket中所有数据，属于危险操作，请谨慎使用。

- 删除单个Object

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

- 删除指定前缀的所有Object

配合-r选项，可以递归执行删除操作，删除指定前缀的所有Object。

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

- 批量删除符合条件的文件

您可以使用--include/--exclude参数，批量删除符合条件的Object。--include/--exclude参数详情请参见[过滤选项](#)。

- 删除所有文件格式不为jpg的Object

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

- 删除所有文件名包含abc且不是jpg和txt格式的Object

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

- 删除指定的Object下未完成的Multipart上传任务的ID

使用加-m选项的rm命令来删除指定未完成的Multipart上传任务的ID。

```
./ossutil rm -m oss://bucket1/obj1/test.txt  
Succeed: Total 1 uploadIds. Removed 1 uploadIds.  
0.900715(s) elapsed
```

- 删除对应前缀的所有未完成的Multipart上传任务的ID

使用加-m和-r选项的rm命令来删除对应前缀的所有未完成的Multipart上传任务的ID。

```
./ossutil rm -m oss://bucket1/ob -r  
Do you really mean to remove recursively multipart uploadIds of oss:  
bucket1/ob(y or N)? y  
Succeed: Total 4 uploadIds. Removed 4 uploadIds.  
1.922915(s) elapsed
```

- 删除对应前缀的所有已上传完成的Object和未完成的Multipart上传任务的ID

使用加-a和-r选项的rm命令来删除对应前缀的所有已上传完成的Object和未完成的Multipart上传任务的ID。

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

```
Succeed: Total 1 objects, 3 uploadIds. Removed 1 objects, 3 uploadIds.
```

常用选项

使用rm命令加不同的选项可以指定删除不同的内容，常用选项如下：

参数名称	描述
-r, --recursive	递归进行操作。当指定该选项时，命令会对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。
-b, --bucket	对Bucket进行操作，该选项用于确认操作作用于Bucket。
-m, --multipart	指定操作的对象为Bucket中未完成的Multipart事件，而非默认情况下的Object。
-a, --all-type	指定操作的对象为Bucket中的Object和未完成的Multipart事件。
-f, --force	强制操作，不进行询问提示。
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--retry-times=	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none"> · info：输出提示信息日志。 · debug：输出详细信息日志（包括http请求和响应信息）。
--include	包含对象匹配模式，如：*.jpg。
--exclude	不包含对象匹配模式，如：*.txt。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.28 set-acl

set-acl用于设置存储空间（Bucket）或对象（Object）的访问权限（ACL）。

命令格式

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

使用示例

- 设置Bucket的ACL

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



说明:

Bucket的ACL可配置为:

- private: 私有
- public-read: 公共读
- public-read-write: 公共读写

ACL详情请参见[#unique_104](#)。

- 设置指定Object的ACL

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



说明:

Object的ACL可配置为:

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

- 设置指定前缀的所有Object的ACL

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

- 批量设置指定条件的Object的ACL

您可以使用--include/--exclude，在设置ACL时选定符合条件的Object。详情请参见[过滤选项](#)。

- 将所有文件格式不为jpg的Object读写权限设置为私有

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

- 将所有文件名包含abc且不是jpg和txt格式的Object读写权限设置为私有

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

常用参数

使用set-acl命令加不同的选项可以针对不同的目标设置ACL，常用选项如下：

参数名称	描述
-r, --recursive	递归进行操作。当指定该选项时，命令会对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。
-b, --bucket	对Bucket进行操作，该选项用于确认操作作用于Bucket。
-f, --force	强制操作，不进行询问提示。
--include	包含对象匹配模式，如：*.jpg。
--exclude	不包含对象匹配模式，如：*.txt。
-j, --jobs	多文件操作时的并发任务数，默认值：3，取值范围：1-10000
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--output-dir	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（关于report文件更多信息，请参考cp命令帮助）。默认值为：当前目录下的ossutil_output目录。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。

参数名称	描述
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***.3128、socks5://120.79.***:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.29 set-meta

set-meta命令用于设置已上传对象（Object）的元信息。

命令格式

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

使用示例

- 设置指定Object的meta信息

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

未指定--update选项和--delete选项时，此命令将用指定的meta代替原来的meta。若未指定[header:value#header:value...]，对于不可删除的Header（不以X-Oss-meta开头的Header），其值不变；其他meta信息将会被删除。



说明:

Header不区分大小写，但Value区分大小写。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
```

更多详情请参见[#unique_120](#)。

- 设置指定前缀的所有Object的meta信息

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

指定了-r选项，将会批量设置指定前缀的Object的meta信息。当一个Object操作出现错误时，会将出错Object的错误信息记录到report文件，并继续操作其他Object，成功操作的Object信息将不会被记录到report文件中。多个meta信息以井号（#）隔开。

- 更新指定Object的meta信息

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

仅更新指定Object的指定Header为输入的value值，其中value可以为空。指定Object的其他meta信息不会改变。--update（可简写为-u）和--delete选项不能同时指定。

- 删除指定Object的meta信息

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

指定--delete选项时，ossutil会删除指定Object的指定Header，此时value必须为空，指定Object的其他meta信息不会改变。对于不可删除的Headers（不以X-Oss-Meta开头的header），该选项不起作用。--update和--delete选项不能同时指定

- 批量设置指定条件的Object的meta信息

您可以使用--include/--exclude，在设置meta信息时选定符合条件的Object。详情请参见[过滤选项](#)。

- 将所有文件类型为jpg的Object设置为低频存储

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

- 将所有Object名包含abc且文件类型不为jpg和txt的Object设置为标准存储

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

常用选项

您可以在使用set-meta命令的时候指定如下选项：

选项名称	描述
-u, --update	用于更新meta信息。
--delete	用于删除指定的meta信息。

选项名称	描述
-r, --recursive	递归进行操作。当指定该选项时，命令会对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。
--include	包含对象匹配模式，如：*.jpg。
--exclude	不包含对象匹配模式，如：*.txt。
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
-j, --jobs	多文件操作时的并发任务数，默认值：3，取值范围：1-10000。
-L, --language	设置ossutil工具的语言，默认值：CH，取值范围：CH/EN，若设置成“CH”，请确保您的系统编码为UTF-8。
--output-dir	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（关于report文件更多信息，请参考cp命令帮助）。默认值为：当前目录下的ossutil_output目录。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none"> · info：输出提示信息日志。 · debug：输出详细信息日志（包括http请求和响应信息）。
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***.3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.30 sign

sign命令用于生成经过签名的url供第三方用户访问存储空间（Bucket）内的对象（Object）。



说明：

关于文件URL的介绍请参见[上传Object后如何获取访问URL](#)。

命令格式

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

- `--timeout`: 设置文件URL的超时时间。单位为秒，默认值为60，取值范围为0-9223372036854775807。
- `--traffic-limit`: 指定http的访问速度。单位: bit/s, 缺省值为0（不受限制），取值范围为819200-838860800，即100KB/s-100MB/s。

使用示例

- 生成默认超时时间的文件URL，默认超时时间为60秒

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

- 生成指定超时时间的文件URL

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

- 生成指定访问速度的文件URL为1MByte/s

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



说明:

单位换算公式: 1MByte=1024KByte=1048576Byte=8388608bit

常用选项

您可以在使用sign命令时附加如下选项:

选项名称	描述
<code>--timeout</code>	设置文件URL的超时时间。单位为秒，默认值为60，取值范围为0-9223372036854775807。
<code>--encoding-type</code>	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
<code>--loglevel</code>	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· <code>info</code>: 输出提示信息日志。· <code>debug</code>: 输出详细信息日志（包括http请求和响应信息）。
<code>--traffic-limit</code>	指定http的访问速度。单位: bit/s, 缺省值为0（不受限制），取值范围为819200-838860800，即100KB/s-100MB/s。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.31 stat

stat命令用于获取指定存储空间（Bucket）或者对象（Object）的描述信息。例如，通过set-meta命令设置的Object元信息，可以通过该命令查看。

命令格式

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

使用示例

- 查看Bucket信息

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

- 查看指定Object信息

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

- 查看名称中含特殊字符的Object信息

ossutil目前只支持以url编码的方式输出或输入Object的文件名，对于无法输入或识别的特殊字符，可以先进行url编码之后再输出。例如，想查看名为`##.txt`的Object信息：

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



注意:

Bucket名称不支持url编码。

常用选项

您可以在使用stat命令时，附加如下选项：

选项名称	描述
--encoding-type	输入或者输出的Object名的编码方式，目前只支持url编码，即指定该选项时，取值为url。如果不指定该选项，则表示Object名未经过编码。Bucket名不支持url编码。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。

选项名称	描述
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--payer	请求的支付方式，如果为请求者付费模式，需将该值设置为requester。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.3.32 update

update命令用于更新ossutil版本。

命令格式

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

该命令检查当前ossutil的版本与最新版本，输出两者的版本号，如果有更新版本，询问是否进行升级。如果指定了--force（简写为-f）选项，当有可用更新时不再询问，直接升级。

使用示例

升级ossutil版本：

```
./ossutil update
当前版本为: v1.5.1, 最新版本为: 1.6.0
确定更新版本(y or N)? y
更新成功!
```

常用选项

您可以在使用update命令时附加如下选项：

选项名称	描述
-f, --force	强制操作，不进行询问提示。
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。

选项名称	描述
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.**.**:3128、socks5://120.79.**.**:1080。
--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。



说明:

更多通用选项请参见[#unique_69](#)。

3.3.33 website

website命令用于添加、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。



说明:

- 静态网站托管功能介绍请参见[#unique_125](#)。
- 回源功能介绍请参考[#unique_126](#)。

命令格式

- 添加/修改website配置

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

ossutil从配置文件`local_xml_file`中读取website配置，并在Bucket中添加对应规则。若Bucket已有website配置，新的配置将覆盖原有配置。



说明:

`local_xml_file`是一个xml格式的文件，其中：

- IndexDocument字段用来配置静态页面默认首页。
- ErrorDocument字段用来配置静态页面默认404页。
- RoutingRules字段用来配置重定向和镜像回源规则。

您可以选择部分规则进行配置，完整配置举例如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>error.html</Key>
```

```

</ErrorDocument>
<RoutingRules>
  <RoutingRule>
    <RuleNumber>1</RuleNumber>
    <Condition>
      <KeyPrefixEquals>abc</KeyPrefixEquals>
      <HttpErrorCodeReturnedEquals>404</HttpErrorC
odeReturnedEquals>
    </Condition>
    <Redirect>
      <RedirectType>Mirror</RedirectType>
      <PassQueryString>true</PassQueryString>
      <MirrorURL>http://www.test.com</MirrorURL>
      <MirrorPassQueryString>true</MirrorPassQueryString>
    >
      <MirrorFollowRedirect>true</MirrorFollowRedirect>
      <MirrorCheckMd5>false</MirrorCheckMd5>
      <MirrorHeaders>
        <PassAll>true</PassAll>
        <Pass>myheader-key1</Pass>
        <Pass>myheader-key2</Pass>
        <Remove>myheader-key3</Remove>
        <Remove>myheader-key4</Remove>
        <Set>
          <Key>myheader-key5</Key>
          <Value>myheader-value5</Value>
        </Set>
      </MirrorHeaders>
    </Redirect>
  </RoutingRule>
  <RoutingRule>
    <RuleNumber>2</RuleNumber>
    <Condition>
      <KeyPrefixEquals>abc</KeyPrefixEquals>
      <HttpErrorCodeReturnedEquals>404</HttpErrorC
odeReturnedEquals>
      <IncludeHeader>
        <Key>host</Key>
        <Equals>test.oss-cn-beijing-internal.aliyuncs.com
</Equals>
      </IncludeHeader>
    </Condition>
    <Redirect>
      <RedirectType>AliCDN</RedirectType>
      <Protocol>http</Protocol>
      <HostName>www.test.com</HostName>
      <PassQueryString>false</PassQueryString>
      <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith
    >
      <HttpRedirectCode>301</HttpRedirectCode>
    </Redirect>
  </RoutingRule>
</RoutingRules>
</WebsiteConfiguration>

```

- 获取website配置

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

`local_xml_file`为文件路径参数。若填写，则将website的配置保存为本地文件；若置空，则将website的配置输出到屏幕上。

- 删除website配置

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

使用示例

- 添加静态网站托管的默认首页和默认404页配置

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

website.xml文件内容为:

```
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>test.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>errortest.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

- 添加镜像回源规则配置

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

website.xml文件内容为:

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

```
</WebsiteConfiguration>
```

- 获取website配置

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

- 删除website配置

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

常用选项

您可以在使用website命令时附加如下选项：

选项名称	描述
--method	表示http的请求类型。取值： <ul style="list-style-type: none">· put：添加或修改配置。· get：获取配置。· delete：删除配置。
--loglevel	设置日志级别，默认为空，表示不输出日志文件。可选值为： <ul style="list-style-type: none">· info：输出提示信息日志。· debug：输出详细信息日志（包括http请求和响应信息）。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。



说明：

更多通用选项请参见[#unique_69](#)。

3.4 查看选项

您可以通过-h选项来查看ossutil支持的所有options。

命令格式

```
./ossutil -h
```

若要查看某个命令支持哪些选项，请使用ossutil help [command]命令来查看，例如ossutil help cp。

通用选项

下面介绍一些ossutil的通用选项，这些选项可以在大部分命令中使用：

- `-c, --config-file`

ossutil工具的配置文件路径，ossutil启动时从配置文件读取配置，在config命令中，ossutil将配置写入该文件。当您需要管理多个账号下的存储空间（Bucket）时，可以生成多个配置文件，其中一个作为默认配置文件。当您管理其他账户下的Bucket时，通过-c指定正确的配置文件即可。

- `-e, --endpoint`

用于指定Bucket的Endpoint，当您需要管理多个地域的Bucket时，可以通过-e选项指定地域。

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

用于指定访问OSS使用的AccessKeyID，当您需要管理多个账号下的Bucket时，可通过-i选项指定正确的AccessKeyID。

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

用于指定访问OSS使用的AccessKeySecret，当您需要管理多个账号下的Bucket时，可通过-k选项指定正确的AccessKeySecret。

- --loglevel

在当前工作目录下输出ossutil日志文件ossutil.log。该值默认为空，表示不输出日志文件，可选值为info/debug。

- 默认级别：该参数默认为空，表示不输出日志文件。
- info级别：用于打印ossutil操作记录。

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

- debug级别：可以输出http流水日志以及原始签名串信息，方便定位问题。

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

- --proxy-host、--proxy-user、--proxy-pwd

当您处于代理上网的环境中，需使用这三个选项指定代理服务器信息。ossutil将使用您指定的信息去通过代理服务器访问OSS。例如：

```
./ossutil ls oss://bucket1 --proxy-host http://47.88.**.**:3128 --  
proxy-user test --proxy-pwd test
```

选项列表

ossutil所有选项列表如下：

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。
--bigfile-threshold	开启大文件断点续传的文件大小阈值，单位为Byte，默认值：100MByte，取值范围：0-9223372036854775807（Byte）。
--acl	配置ACL信息。
--range	下载文件时，指定文件下载的范围，格式为：3-9或3-或-9。
--type	计算的类型，默认值：crc64，取值范围：crc64/md5。
-v, --version	显示ossutil的版本并退出。
-u, --update	更新操作。
--origin	表示http请求头origin字段的值。
--upmode	表示上传模式，缺省值为normal，取值为：normal、append、multipart，分别表示正常上传、追加上传、分块上传，主要在probe命令中使用。
--sse-algorithm	表示服务端加密算法，取值为KMS或AES256。
--include	包含对象匹配模式，如：*.jpg。
--exclude	不包含对象匹配模式，如：*.txt。

选项名称	描述
-r, --recursive	递归进行操作。对于支持该选项的命令，当指定该选项时，命令会对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。
--addr	表示一个网络地址，通常为域名，主要在probe命令中使用。
--kms-masterkey-id	表示KMS密钥托管服务中的主密钥ID。
-m, --multipart	指定操作的对象为Bucket中未完成的Multipart事件，而非默认情况下的Object。
-d, --directory	返回当前目录下的文件和子目录，而非递归显示所有子目录下的所有Object。
--payer	请求的支付方式，如果为请求者付费模式，可以将该值设置成“requester”。
--maxupspeed	最大上传速度，单位：KB/s，缺省值为0（不受限制）。
--retry-times	当错误发生时的重试次数，默认值：10，取值范围：1-500。
-c, --config-file	ossutil工具的配置文件路径，ossutil启动时从配置文件读取配置，在config命令中，ossutil将配置写入该文件。
--download	表示从OSS下载，主要在probe命令中使用。
-j, --jobs	多文件操作时的并发任务数，默认值：3，取值范围：1-10000。
-a, --all-type	指定操作的对象为Bucket中的Object和未完成的Multipart事件。
--disable-empty-referer	表示不允许referer字段为空，主要在referer命令中使用。
--method	表示http的请求类型，取值为PUT、GET、DELETE等。
--output-dir	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（关于report文件更多信息，请参考cp命令帮助）。默认值为：当前目录下的ossutil_output目录。
--meta	设置Object的meta为[header: value#header: value...], 如：Cache-Control: no-cache#Content-Encoding: gzip。
--object	表示OSS中对象的名称，主要在命令probe中使用。
-e, --endpoint	ossutil工具的基本endpoint配置（该选项值会覆盖配置文件中的相应设置），注意其必须为一个二级域名。
--limited-num	返回结果的最大个数。
-L, --language	设置ossutil工具的语言，默认值：CH，取值范围：CH/EN，若设置成“CH”，请确保您的系统编码为UTF-8。
--delete	删除操作。

选项名称	描述
-b, --bucket	对Bucket进行操作，该选项用于确认操作作用于Bucket。
--disable-crc64	该选项关闭crc64，默认情况下，ossutil进行数据传输都打开crc64校验。
--upload	表示上传到OSS，主要在probe命令中使用。
--part-size	分片大小，单位为Byte，默认情况下ossutil根据文件大小自行计算合适的分片大小值。如果有特殊需求或者需要性能调优，可以设置该值，取值范围：1-9223372036854775807（Byte）。
--timeout	签名url的超时时间，单位为秒，默认值为：60，取值范围：0-9223372036854775807。
-k, --access-key-secret	访问OSS使用的AccessKeySecret（该选项值会覆盖配置文件中的相应设置）。
--checkpoint-dir	checkpoint目录的路径（默认值为：.ossutil_checkpoint），断点续传时，操作失败ossutil会自动创建该目录，并在该目录下记录checkpoint信息，操作成功会删除该目录。如果指定了该选项，请确保所指定的目录可以被删除。
--url	表示一个url地址，主要在probe命令中使用。
--marker	列举Bucket时的marker，或列举Object或Multipart Upload时的key marker。
-f, --force	强制操作，不进行询问提示。
--snapshot-path	该选项用于在某些场景下加速增量上传批量文件或者增量下载批量Object。上传文件或者下载Object时使用该选项，ossutil在指定的目录下生成快照文件，记录文件上传或者Object下载的快照信息，在下次指定该选项上传或下载时，ossutil会读取指定目录下的快照信息进行增量上传或者下载。用户指定的snapshot目录必须为本地文件系统上的可写目录，若该目录不存在，ossutil会创建该文件用于记录快照信息，如果该目录已存在，ossutil会读取里面的快照信息，根据快照信息进行增量上传（只上传上次未成功上传的文件和本地进行过修改的文件）或者增量下载（只下载上次未成功下载的Object和修改过的Object），并更新快照信息。注意：该选项在本地记录了成功上传的文件的本地lastModifiedTime或者记录了下载Object的lastModifiedTime，从而在下次上传或者下载时通过比较lastModifiedTime来决定是否跳过相同文件的上传或者跳过相同的Object下载。当使用该选项上传时，请确保两次上传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，如果想要增量上传批量文件，请使用--update选项。ossutil不会主动删除snapshot-path下的快照信息，当用户确定快照信息无用时，请用户及时自行删除snapshot-path。

选项名称	描述
--loglevel	<p>设置日志级别，默认为空，表示不输出日志文件，可选值为： info、debug。</p> <ul style="list-style-type: none"> · info：输出提示信息日志。 · debug：输出详细信息日志（包括http请求和响应信息）。
--storage-class	设置对象的存储方式，默认值：Standard，取值范围：Standard、IA、Archive。
-i, --access-key-id	访问OSS使用的AccessKeyID（该选项值会覆盖配置文件中的相应设置）。
-t, --sts-token	访问OSS使用的STSToken（该选项值会覆盖配置文件中的相应设置），非必须设置项。
--parallel	单文件内部操作的并发任务数，取值范围：1-10000，默认将由ossutil根据操作类型和文件大小自行决定。
--partition-download	<p>分区下载使用，一个ossutil命令下载一个分区，其值格式为“分区编号：总分区数”，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；利用该选项，待下载的Object分成多个区，可以由多个ossutil命令一起下载完成，每个ossutil命令下载各自的分区，多个ossutil命令可以并行在不同机器上执行。</p>
--bucketname	表示Bucket的名称，主要在probe命令中使用。
--encoding-type	<p>输入或者输出的Object名或文件名的编码方式，目前只支持url encode，即指定该选项时，取值为url，如果不指定该选项，则表示Object名或文件名未经过编码。Bucket名不支持url encode。</p> <div>  注意： 如果指定了该选项，则形如oss: //bucket/object的cloud_url，输入形式为：oss: //bucket/url_encode (object)，其中oss: //bucket/字符串不需要编码。 </div>
--origin	http请求头Origin字段的值，表示请求来源域，用来标示跨域请求。
--acr-method	http请求头Access-Control-Request-Method的值，可选值为：GET、PUT、POST、DELETE、HEAD。
--acr-headers	http请求头Access-Control-Request-Headers的值，表示在实际请求中会用到的除简单头部之外的头。如果有多个取值，各个header用英文的逗号（,）分隔，再加上双引号。例如：--acr-headers "header1,header2,header3"。
--upload-id-marker	列举Multipart Uploads时的uploadID marker。

选项名称	描述
-h, --help	显示帮助信息。
--proxy-host	网络代理服务器的url地址，支持http、https、socks5。例如http://120.79.***:3128、socks5://120.79.***:1080。
--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。
--traffic-limit	指定http的访问速度。单位：bit/s，缺省值为0（不受限制），取值范围为819200-838860800，即100KB/s-100MB/s。用于sign命令。

3.5 常见问题

本文介绍在使用 ossutil 时的常见问题及处理方法。

使用 -u 参数上传文件时出现 skip 的情况

使用 -u 参数上传文件的时候，ossutil 会将上传文件和 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 是正常现象。

文件递归解冻时出现 403 报错

问题分析：操作解冻文件的过程中出现 403，有两种可能性：

- 用户使用子账号操作文件，权限不够。
- 用户的文件是违禁内容被封禁掉了。

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。

文件递归解冻时出现 400 报错

排查步骤：

1. 检查您的命令和官方提供的命令是否完全一致，避免误操作。
2. 使用 stat 选项看一下文件的状态是否是已经解冻状态。若文件已经解冻，再次操作时就会出现 400 报错。

访问OSS时出现签名错误 (signature not match)

排查步骤:

1. 在所有的操作命令后加上 `--loglevel=debug` 选项, 将请求的 loglevel 设置为 debug 级别, 可以将签名前字符串与 http 流水信息打印到日志文件 ossutil.log。

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

2. 根据官网的[签名规则](#), 将签名前后的日志信息进行对比, 定位问题。
3. 若无法解决, 请联系[技术支持](#)。

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。

ossimport具有以下特点：

- 支持丰富的数据源，包括本地、七牛、百度BOS、AWS S3、Azure Blob、又拍云、腾讯云COS、金山KS3、HTTP、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。
- 支持迁移指定时间后的文件以及特定前缀的文件。
- 支持并行数据下载和上传。



说明：

- 您也可以使用[在线迁移服务](#)迁移您的数据，无需再部署迁移工具。
- 对于小于30TB的数据从本地文件或可以挂载到本地文件系统的迁移情况，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及以上版本



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30TB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。[下载地址](#)
- 分布式模式：当您需要迁移的数据大于30TB时，推荐使用分布式模式。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。[下载地址](#)



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。

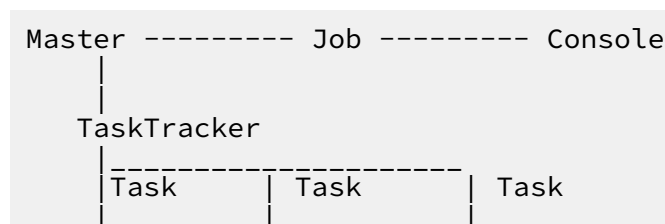
单机模式下文件结构如下：

```
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：



Worker	Worker	Worker
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： <ol style="list-style-type: none"> 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	<ul style="list-style-type: none"> · 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。	
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job.cfg。	
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
<code>workingDir</code>	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。
<code>workerUser</code>	Worker机器的ssh用户名	<ul style="list-style-type: none"> - 如果配置了<code>privateKeyFile</code>，则优先使用<code>privateKeyFile</code>。 - 如果没有配置<code>privateKeyFile</code>，则使用<code>workerUser/workerPassword</code>。 - 单机模式不需要修改此参数。
<code>workerPassword</code>	Worker机器的ssh用户密码	单机模式不需要修改此参数。
<code>privateKeyFile</code>	private key文件路径	<ul style="list-style-type: none"> - 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<code>privateKeyFile</code>，则优先使用<code>privateKeyFile</code>。 - 如果没有配置<code>privateKeyFile</code>，则使用<code>workerUser/workerPassword</code>。 - 单机模式不需要修改此参数。
<code>sshPort</code>	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。
<code>workerTaskThreadNum</code>	Worker执行Task的最大线程数	<ul style="list-style-type: none"> - 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。
<code>workerMaxThrougput(KB/s)</code>	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。
<code>dispatcherThreadNum</code>	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。

参数	含义	说明
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。	<ul style="list-style-type: none"> - 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import或audit，默认 import。 <ul style="list-style-type: none"> - import，执行数据迁移操作并校验迁移数据的一致性。 - audit，仅进行数据一致性校验。
isIncremental	是否打开增量迁移模式，布尔类型。	<ul style="list-style-type: none"> - 默认值false。 - 如果设为true，会每间隔incrementalModeInterval（单位秒）重新扫描一次增量数据，并将增量数据迁移到OSS。
incrementalModeInterval	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，不建议配置成大于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。	<ul style="list-style-type: none"> - 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。

参数	含义	说明
srcType	同步源类型，字符串，请注意大小写。	支持以下类型： <ul style="list-style-type: none"> - local：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。 - oss：从一个 OSS bucket 迁移到另一个 bucket。 - qiniu：从七牛云存储迁移到OSS。 - bos：从百度的云存储迁移到OSS。 - ks3：从金山云存储迁移到OSS。 - s3：从 AWS S3 迁移到OSS。 - youpai：从又拍云迁移到OSS。 - http：通过提供的HTTP链接列表迁移数据到OSS。 - cos：从腾讯云存储COS迁移到OSS。 - azure：从Azuer Blob迁移到OSS。
srcAccessKey	源AccessKey，字符串。	<ul style="list-style-type: none"> - 如果srcType设置为oss、qiniu、baidu、ks3、s3，则填写数据源的AccessKey。 - 如果srcType设置为local、http，则该项不需要填写。 - 如果srcType设置为youpai、azure，则填写用户名AccountName。
srcSecretKey	源SecretKey，字符串。	<ul style="list-style-type: none"> - 如果srcType设置为oss、qiniu、baidu、ks3、s3，则填写数据源的 SecretKey。 - 如果srcType设置为local、http，则该项不需要填写。 - 如果srcType设置为youpai，则填写操作员密码。 - 如果srcType设置为azure，则填写 AccountKey。

参数	含义	说明
srcDomain	源Endpoint	<ul style="list-style-type: none"> - 如果srcType设置为local、http，则该项不需要填写。 - 如果srcType设置为oss，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qiniu，则填写从七牛控制台获取的对应bucket的域名。 - 如果srcType设置为bos，则填写百度BOS域名，如http://bj.bcebos.com 或 http://gz.bcebos.com。 - 如果srcType设置为ks3，则填写金山ks3域名，如http://kss.ksyun.com或http://ks3-cn-beijing.ksyun.com或http://ks3-us-west-1.ksyun.com。 - 如果srcType设置为S3，则填写AWS S3各region的域名。 - 如果srcType设置为youpai，则填写又拍云域名，如自动判断最优线路http://v0.api.upyun.com或电信线路http://v1.api.upyun.com或联通网通线路http://v2.api.upyun.com或移动铁通线路http://v3.api.upyun.com。 - 如果srcType设置为cos，则填写腾讯云bucket所在的区域，例如华南园区:gz、华北园区:tj、华东园区:sh。 - 如果srcType设置为azure，则填写Azure Blob连接字符串中的EndpointSuffix，如core.chinacloudapi.cn。
srcBucket	源bucket名字或container名称	<ul style="list-style-type: none"> - 如果srcType设置为local、http，则不需要填写。 - 如果srcType设置为azure，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	<ul style="list-style-type: none"> - 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c:/example/ 或者/data/example/ 的格式。 <div>  注意： c:/example//或 /data//example/ 或 /data/example//是非法的。 </div> <ul style="list-style-type: none"> - 如果srcType设置为oss、qiniu、bos、ks3、youpai、s3，则填写待同步object的前缀，不包括bucket名称，如data/to/oss/。 - 如需同步所有文件，则srcPrefix设置为空。
destAccessKey	目的AccessKey，字符串。	OSS的AccessKeyID，请到 阿里云控制台 查看。
destSecretKey	目的SecretKey，字符串。	OSS的AccessKeySecret，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。

参数	含义	说明
destPrefix	目标前缀，字符串，默认为空。	<ul style="list-style-type: none"> - 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oss的某个目录下，请以/结尾，如data/in/oss/。 - 注意oss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srcPrefix+relativePath的文件，迁移到oss的路径为destDomain/destBucket/destPrefix + relativePath。 - 一个云端文件路径为srcDomain/srcBucket/srcPrefix+relativePath的文件，迁移到oss的路径为destDomain/destBucket/destPrefix+relativePath。
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	<ul style="list-style-type: none"> - 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=cos时有效。

参数	含义	说明
httpListFilePath	HTTP列表文件的绝对路径，字符串。	<ul style="list-style-type: none"> - 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	job.cfg	srcPrefix是以正斜线 (/) 结尾的绝对路径，如 D:/work/oss/data/、/home/user/work/oss/data/。
从七牛云存储迁移到OSS	job.cfg	srcPrefix和destPrefix可以配置为空；如果不为空，请以正斜线 (/) 结尾，如 destPrefix=docs/。
从百度bos迁移到OSS	job.cfg	srcPrefix和destPrefix可以配置为空；如果不为空，请以正斜线 (/) 结尾，如destPrefix=docs/。
从AWS S3迁移到OSS	job.cfg	参见 S3域名列表 。

迁移类型	配置文件	说明
从又拍云存储迁移到OSS	job.cfg	srcAccessKey/srcSecretKey填操作员账号及密码。
从腾讯cos迁移到OSS	job.cfg	srcDomain请按照V4版本填写，如srcDomain=sh。srcPrefix可以为空，当不为空时候，请以 / 开头和结尾，如srcPrefix=/docs/。
从Azure blob迁移到OSS	job.cfg	srcAccessKey/srcSecretKey填存储账号及密钥。srcDomain填连接字符串中的EndpointSuffix，如core.chinacloudapi.cn。
从OSS迁移到OSS	job.cfg	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。



说明:

- 配置文件参数请尽量在启动迁移前确定。
- `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`incrementalModeInterval`，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

本文介绍如何单击部署ossimport，单机部署支持Linux和Windows系统。

下载

单机版本下载地址[ossimport-2.3.4.zip](#)，下载到本地后，使用工具或命令unzip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

配置

单机版本有两个配置文件`conf/sys.properties`、`conf/local_job.cfg`。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`。

其它配置项请按照实际需求配置。



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。

运行

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。
- 分步执行：执行启动服务、提交任务、重试失败子任务等步骤。



说明：

对于初级用户建议使用一键导入。

· 一键导入

1. 执行一键导入，Window系统下在`cmd.exe`中执行`import.bat`，Linux终端中执行 `bash import.sh`。
2. 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。
3. 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。
4. 当Job完成后，如果发现有任务失败了，会提示是否重试，输入y重试，输入n则跳过退出。
5. 如果上传失败，请打开文件`master/jobs/local_test/failed_tasks/<tasktaskid>/audit.log`查看，确定失败原因。

· 分步执行



说明：

没有特殊需要，请使用 一键导入方式迁移数据。

1. 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`。

2. 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`。

3. 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`。

4. 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`。

5. 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`。

6. 停止服务。

Windows下关闭窗口`%JAVA_HOME%/bin/java.exe bash console.sh stop`。


· 常见失败原因

- 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。

常见错误及排除

请查看[常见错误及排除](#)。

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

分布式版本下载地址[ossimport-2.3.4.tar.gz](#)，下载到本地后，使用命令 `tar -zxvf ossimport-2.3.4.tar.gz -C $HOME/ossimport` 解压，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
```

注意：

- `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，需要从新执行任务，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。



说明：

配置项提交后不能修改，请在提交前确定。

· 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
│       ├── xxttooss   # 任务名称
│       └── checkpoints # Master分解Job到Task的checkpoint点记
└── 录
    └── 0
        └── ED09636A6EA24A292460866AFDD7A89A.cpt
```

192.168.1.6 错误原因表	└─ 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@
	└─ audit.log	# Task运行日志，通过该日志可以查看
		└─ DONE
	└─ error.list	# Task运行成功标志，失败为空
		# Task错误列表，可以查看错误文件列
	└─ 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
192.168.1.6 误原因	└─ TASK	# Task描述信息
		└─ pending_tasks
	└─ succeed_tasks	# 尚未分配的Task
		# 成功运行的Task
	└─ audit.log	# Task运行日志，通过该日志可以查看错
		└─ DONE
	└─ error.list	# Task运行成功标志
		# 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
└─ worker	└─ TASK	# Task描述信息
		# 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。

- 1. UnsupportedClassVersionError异常

执行命令时异常：

```
Exception in thread "main" java.lang.UnsupportedClassVersionError:
com.aliyun/ossimport2/OSSImport2 : Unsupported major.minor version
51.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClassCond(ClassLoader.java:
631)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:615)
    at com.simontuffs.onejar.JarClassLoader.defineClass(
JarClassLoader.java:693)
    at com.simontuffs.onejar.JarClassLoader.findClass(JarClassLo
ader.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)
```

原因：Java版本过低，更新到1.7或以上版本。

- 2. InvocationTargetException异常

使用submit命令提交任务报异常：

```
Exception in thread "main" java.lang.reflect.InvocationTargetExce
ption
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native
Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMeth
odAccessorImpl.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
```

原因：检查是否将配置文件中原有的项删除或者注释掉了，不需要配置的项请在等号后面填空，不需要删除。

- 3. too many open files

原因：`ulimit -n` 查看系统句柄。

- 如果值小于一万以下的，可以通过`ulimit -n 65536`调大后，重启进程即可；
- 如果本来就设置的比较大，那就用`sudo lsof -n`排查是哪些进程打开了句柄。

- 4. windows启动后秒退

原因：大部分情况是java没装或者版本低于1.7导致的，或配置文件错误导致。

- 5. no jobs is running or finished

`submit`命令提交完任务后，用`stat`查看任务状态一直显示：

```
bash console.sh stat
[WARN] List files dir not exist : /home/<user>/ossimport/workdir/
master/jobs/
no jobs is running or finished.
```

原因：

- Job 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，打印该日志是正常的；
- 很长一段时间后依旧打印该错误，一般是没有使用`start`命令启动进程或者进程启动后有异常退出。没有启动服务的情况，只需使用`start`就可以了；否则看下`logs/ossimport.log`，找到异常原因并解决，然后再启动服务进程。

- 6. stat命令一直显示scanFinished: false

观察Task的总数是否在增加：

- 如果Task总数增加，则表明Job的文件列表还在List新的文件。
- 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，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数也不会增多，检查日志是否有异常。

- 7. 服务进程挂掉，日志却没有输出异常

原因：如果机器的可用内存是少于2GB 的，大概率是内存不足被杀掉了，检查下`dmsg`日志是否有内存不足被杀的记录。

- 8. 进程挂掉或者杀掉进程后，重启服务需要做什么操作

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，只要不调用`clean`命令，所有提交过的Job都有断点记录，不会重做已经完成的工作。

- 9. 任务完成OSS控制台显示的数据量比源数据量小

Job全部成功上传完后OSS控制台里没有看到bucket的大小有变化，或者和本地用du统计的大小相差很大。原因：OSS控制台的bucket数据量是延迟1小时更新的。du命令统计的是块大小，会比实际文件要大，可以参考如下命令统计本地目录的真实大小：ls -lR <目录绝对路径> | grep "\-rw" | awk '{sum+=\$5}END{print sum}'。

- 10. 如何处理stat显示的失败任务

```
[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 ] - =====
```

如上图所示，使用stat命令查看迁移任务状态。如果Failed Task Count不为0，则迁移任务失败。迁移完成后请使用retry命令进行重试。

- 11. 有些失败任务后反复retry都无法成功

原因：查看文件\$work_dir/master/jobs/\$jobName/failed_tasks/\$taskName/error.list拿到失败文件的相对路径，检查该文件是否有权访问，是否被删除，是否是软链接，是否是乱码的文件名等。

- 12. 文件名乱码的文件如何上传到OSS

需要首先使用export LANG="<your file name encode>", ls 查看该文件名不乱码之后，用clean命令清掉原来的Job，再用submit命令重新提交Job。

- 13. java.nio.file.AccessDeniedException

报异常：启动服务时报java.nio.file.AccessDeniedException。原因：没有权限访问配置文件目录。

- 14. Task格式显示为0，但JobState显示成功

如下图所示，Pending Task和Diapatched Task 都显示为0，但JobState显示为SUCCEED：

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

```
[2015-12-28 16:12:35] [INFO] JobState:SUCCEED
```

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。
- 15. The bucket you are attempting to access must be addressed using the specified endpoint

日志报异常:

```
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>
```

原因: bucket的`srcDomain`或`destDomain`填写错误, 请按照[域名列表](#)填写正确的域名。

- 16. The request signature we calculated does not match the signature you provided

日志报异常:

```
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
```

原因: 检查`destAccessKey`和`destSecretKey`是否填错, 详细说明请参看[访问控制](#)。

- 17. InvocationTargetException

submit命令提交任务时报异常:

```
submit job:/disk2/ossimport2/local_job.cfg
Exception in thread "main" java.lang.reflect.InvocationTargetExce
ption
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native
Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMeth
odAccessorImpl.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
```

原因：检查conf/sys.properties中的配置项 workingDir 是否配置，配置是否正确；检查配置文件路径是否正确。

· 18. 是否支持设置代理

不支持

· 19. OSS迁移到OSS为什么还会产生费用

参考[访问域名](#)里的域名帮助，配置内网域名后，将不收取流量费用，但访问次数的费用是依旧计费的。

· 20. 同步的过程中显示源端的文件不存在

原因：Master是先 List 出文件列表，之后按照文件列表做数据迁移。当 List 完之后，源端某些文件被删除，就会出现源端文件不存在的情况。这种文件会被跳过，并把文件输出在错误列表里。

· 21. 开启增量模式，本地删除后OSS上会不会被删

开启增量模式，本地删除后OSS上会不会被删，删除操作不会被同步。

· 22. 开启增量模式，有些新增的文件没有被同步

增量模式是采用对比文件最后修改来判断文件是否为增量的，Linux的mv，Windows的cp，mv以及rsync带-t或者-a参数等操作是不会修改文件的最后修改时间的，这些操作修改的文件是不会被扫描到的。

· 23. 又拍迁移的任务数一直显示0

原因：又拍的比较复杂，主要分两种情况：

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

这条日志如果recentLatency=300000这种一般就是在正常 List 的，又拍List 比较慢一般都是跑满30秒超时，30秒 List 出几个文件就返回几个文件过来，这种情况等任务慢慢List 出来就正常了；

- recentLatency很小，这种情况一般就是账号密码填错之类的，因为又拍的sdk出错只返回null，并不返回错误结果，所以只能通过抓包的方式获得又拍返回的错误码排查。

- 24. 又拍迁移时srcAccessKey和srcSecretKey填写什么

填写又拍的操作员账号和密码。

- 25. 又拍迁移时一直显示http 429错误

又拍对sdk访问间隔做了限制，如果访问稍微快一点，会进行限速，请联系又拍的客服放开限制。OssImport本身会对这种情况进行重试的。

- 26. 执行时unknown command “java”， unknown command “nohup”之类的提示

原因：使用的命令没有安装，请用yum或apt-get或zypper 等命令安装相应的命令。

- 27. 任务与配置文件不符

Job配置文件是对的，但跑的时候明显和Job配置文件配置的不一样。只有sys.properties是改完之后重启就生效的，Job的配置文件一旦提交后，修改是不会生效的，需要 clean 掉原有的Job，然后重新submit新的配置文件。

- 28. The bucket name “xxx/xx” is invalid

日志报异常：

```
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.
```

原因：检查配置项destBucket是否填写正确，bucket是不带正斜线 (/) 以及路径的。

- 29. com.aliyun.oss.ClientException: Unknown

日志报异常：

```
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.
```

以及SocketTimeoutException，一般都是网络打满的时出现的，OssImport内部会重试，如果重试完后依旧失败，可以在任务完成后调用retry命令再次重试。

- 30. Connect to xxx.oss-cn-beijing-internal.aliyuncs.com:80 timed out

日志报异常：

```
Unable to execute HTTP request: Connect to xxx.oss-cn-beijing-
internal.aliyuncs.com:80 timed out
[ErrorCode]: ConnectionTimeout
[RequestId]: Unknown
```

原因：非ECS的机器不能使用带internal的域名。

- 31. The specified bucket is not valid

日志报异常：

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

原因：配置文件里的destDomian配置的域名不能带bucket名称。

- 32. 配置文件里的srcPrefix能单独指定文件吗

不可以，srcPrefix只支持目录或者前缀级别，单个文件上传可以用别的更简单的工具。

- 33. Unable to execute HTTP request: The Difference between ... is too large.

日志报异常：

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

原因：

- 本地机器时间不对，与服务器时间相差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。

- 34. No route to host

日志里显示错误No route to host，一般是本地防火墙或者 iptables 等原因导致网络不通。

- 35. Unknown http list file format

使用 http 模式日志显示该错误，是因为指定的http列表文件格式不对：

- 从其它系统上拷过来的文件，可以用mac2unix，dos2unix等相关命令转化文件格式；
- 文件里有某些行不符规则，比如某行少于两列。

- 36. The boject key “/xxxxx.jpg” is invalid

日志报异常：

```
Exception:java.lang.IllegalArgumentException: The boject key "/xxxxx  
.jpg" is invalid. An object name should be between 1 - 1023 bytes
```

```
long when encoded as UTF-8 and cannot contain LF or CR or unsupported chars in XML1.0, and cannot begin with "/" or "\".
```

原因：

- 检查srcPrefix是否是作为目录的但没有以 / 结尾；
- 检查destPrefix是否以 / 或者 \ 开头了。

5 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/删除规则，并自动生成策略的JSON文本。用户添加完所有规则后，只需要将JSON文本拷贝，然后粘贴到访问控制（RAM）控制台的创建授权策略内容框内。

具体操作请参见[创建自定义授权策略](#)。

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，每个是以下形式：

- 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ListObjects）
- 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ListObjects）
- 填写完整的资源路径：`acs:oss:*:1234:my-bucket/dir`，其中1234为用户的User ID（在控制台查看）

EnablePath

当用户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，例如用户对`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，用户还需要以下权限：

```
ListObjects my-bucket
ListObjects my-bucket/users
ListObjects my-bucket/users/dir
```

勾选EnablePath选项时，上面这些权限会自动添加。

· Conditions

指定授权访问时应该满足的条件，可以指定多个。

例子

授权对`my-bucket`及其文件全部的权限：

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/*	

更多例子请参见[RAM Policy Editor](#)。

6 ossftp

6.1 如何快速安装ossftp

ossftp工具是一个特殊的FTP server。它接收普通FTP请求后，将对文件、文件夹的操作映射为对OSS的操作，从而使得您可以基于FTP协议来管理存储在OSS上的文件。

功能简介

- 主要特性
 - 跨平台：无论是Windows、Linux还是Mac，无论是32位还是64位操作系统，无论是图形界面还是命令行都可以运行。
 - 免安装：解压后可直接运行。
 - 免设置：无需设置即可运行。
 - 透明化：FTP工具是Python写的，您可以看到完整的源码，我们稍后也会开源到Github。
- 主要功能
 - 支持文件和文件夹的上传、下载、删除等操作。
 - 通过Multipart方式，分片上传大文件。
 - 支持大部分FTP指令，可以满足日常FTP的使用需求。



说明:

- ossftp工具主要面向个人用户使用，生产环境请使用OSS SDK。
- 在目前的1.0版本中，考虑到安装部署的简便，ossftp工具不支持TLS加密。由于FTP协议是明文传输的，为了防止您的密码泄漏，建议将FTP Server和Client运行在同一台机器上，通过127.0.0.1:port的方式来访问。
- 不支持rename和move操作。
- 安装包解压后的路径不能包含中文。
- FTP Server的管理控制页面在低版本的IE中可能打不开。
- FTP Server支持的Python版本：Python2.6和Python2.7。

下载地址

- Windows: [ossftp-1.0.3-win.zip](#)

Windows默认不会安装Python2.7，所以安装包中包含了Python2.7，解压后即可使用。

- Linux/Mac: [ossftp-1.0.3-linux-mac.zip](#)

Linux/Mac系统默认安装Python2.7或Python2.6，所以安装包中不再包含可执行的Python，只包含了相关依赖库。

快速运行

1. 下载和您系统匹配的安装包。
2. 将安装包解压后运行。

- Windows系统

解压后双击运行`start.vbs`即可。如双击无反应，请升级您的IE浏览器或设置其他浏览器为默认浏览器。

- Linux系统

- a. 解压下载的文件：

```
$ unzip ossftp-1.0.3-linux-mac.zip
```

- b. 进入解压后的文件夹，运行`start.sh`：

```
$ cd ossftp-1.0.3-linux-mac
$ bash start.sh
```

- c. 使用一台可以访问这台服务器的，有图形化界面的电脑，通过浏览器访问FTP服务器操作界面。访问域名：`http://ServerIP:8192`。

- Mac系统

解压后双击`start.command`，或者在终端运行`$ bash start.command`。

3. 上述步骤会启动一个FTP server，默认监听在127.0.0.1的2048端口。为了方便您对FTP server的状态进行管控，还会启动一个web服务器，监听在127.0.0.1的8192端口。

OSS-FTP
配置
日志
关于

系统配置选项

开机自动启动 ☐

启动时弹出状态页 ☒

显示托盘图标 ☒

ossftp 监听地址 (重启生效)

ossftp 监听端口 (重启生效)

ossftp 日志等级 (DEBUG, INFO, WARNING, ERROR, CRITICAL。重启生效)

Bucket endpoints (大部分情况无需设置, 如何设置请参考wiki。重启生效)

Language(cn/en)

保存配置

重启

退出

- ossftp监听地址：填写需要使用FTP服务的客户端IP。如果是在本机上运行客户端，保持默认即可。
- ossftp监听端口：ossftp的监听端口，默认即可。
- ossftp日志等级：ossftp的日志输出等级，根据需求填写。
- Bucket endpoints：填写Bucket的访问域名（格式为bucket_name.endpoint），多个域名以英文逗号（,）隔开。
- Language：选择ossftp的显示语言。

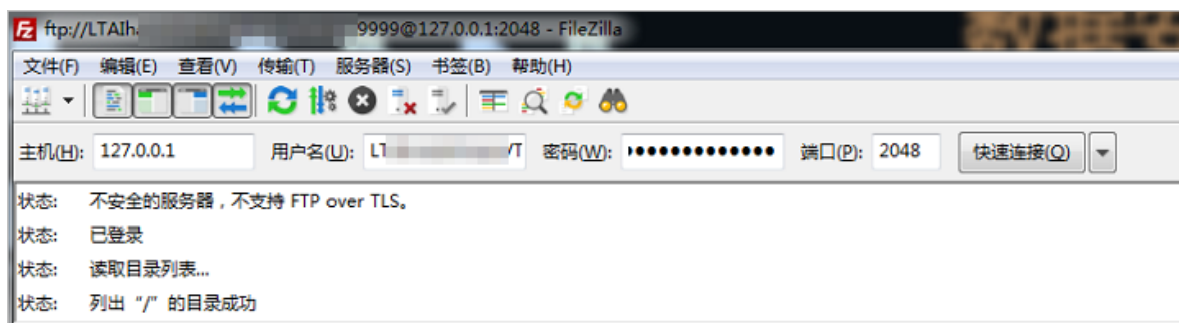


说明:

- 配置完成后需保存配置并重启生效。
- 同一时间内只能存在一个服务器和一个连接。如果在一个服务器已连接的情况下新建连接，则之前连接会直接断开。

4. 下载[FileZilla客户端](#)并安装。

5. 配置OSS访问信息后单击快速连接。



- 主机：配置服务器IP地址，若服务器和客户端在一台设备上，使用默认地址：127.0.0.1。
- 用户名：填写拥有Bucket访问权限的AccessKeyID和Bucket名称，格式为AccessKeyID/bucket_name，例如tSxyi*****wPMEp/test-hz-jh-002。
- 密码：填写拥有Bucket访问权限的AccessKeySecret。
- 端口：填写服务器配置的监听端口，默认填写2048。



说明：

AccessKeyID和AccessKeySecret的获取，请参见[创建RAM用户](#)中的创建AccessKey部分。

6.2 Discuz如何存储远程附件到OSS

本文介绍如何基于Discuz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[#unique_142](#)。
 - 创建Bucket的步骤请参见[#unique_143](#)。
- 已搭建Discuz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Discuz版本为Discuz! X3.1。

配置步骤

1. 使用管理员账号登录Discuz站点。

2. 在管理界面单击全局 > 上传设置。



Discuz! Control Panel

首頁 **全局** 界面 內容 用戶 門戶 論壇 群組 防灌水 運營 應用 工具 站長 UCenter

全局 » 站點信息 [+]

站點信息

站點名稱:
 站點名稱，將顯示在瀏覽器窗口標題等位置

網站名稱:
 網站名稱，將顯示在頁面底部的聯繫方式處

網站 URL:
 網站 URL，將作為鏈接顯示在頁面底部

管理員郵箱:
 管理員 E-mail，將作為系統發郵件的時候的發件人地址

QQ登錄及賬號碼  **2**
 [設置我的QQ在線狀態](#)

網站備案信息代碼:
 頁面底部可以顯示 ICP 備案信息，如果網站已備案，在此輸入您的授權碼，它將顯示在

- 站點信息
- 註冊與訪問控制
- 站點功能
- 性能優化
- SEO設置
- 域名設置
- 廣播設置
- 空間設置
- 用戶權限
- 積分設置
- 時間設置
- 上傳設置**
- 水印設置
- 附件類型尺寸
- 搜索設置
- 地區設置
- 排行榜設置
- 手機版訪問設置
- 防採集設置

3. 单击远程附件，设置远程附件选项。

Discuz! Control Panel

首页 全局 界面 内容 用户 门户 论坛 群组 运营 应用 工具 云平台 站长 UCenter

全局 » 上传设置 [+]

上传设置 基本设置 **远程附件** 论坛附件 空间附件 门户附件

启用远程附件:
☒ 是 ☐ 否 **1**

启用 SSL 连接:
☐ 是 ☒ 否 **2** 注意: FTP 服务器必需开启了 SSL

FTP 服务器地址:
 3 可以是 FTP 服务器的 IP 地址或域名

FTP 服务器端口:
 4 默认为 21

FTP 帐号:
 5 该帐号必需具有以下权限: 读取文件、写入文件、删除文件、创建目录、子目录继承

FTP 密码:
 6 基于安全考虑将只显示 FTP 密码的第一位和最后一位, 中间显示八个 * 号

被动模式(pasv)连接: **7**
☒ 是 ☐ 否 一般情况下非被动模式即可, 如果存在上传失败问题, 可尝试打开此设置

Discuz! Control Panel

首页 全局 界面 内容 用户 门户 论坛 群组 防灌水 运营 应用 工具 站长 UCenter

全局 » 上传设置 [+]

上传设置 基本设置 **远程附件** 论坛附件 空间附件 门户附件

远程附件目录:
 8 远程附件目录的绝对路径或相对于 FTP 主目录的相对路径, 结尾不要加斜杠「/」, 「.»表示 FTP 主目录

远程访问 URL:
 9 支持 HTTP 和 FTP 协议, 结尾不要加斜杠「/」; 如果使用 FTP 协议, FTP 服务器必需支持 PASV 模式, 有限

FTP 传输超时时间:
 10 单位: 秒, 0 为服务器默认

连接测试:
 11 无需保存设置即可测试, 请在测试通过后再保存

允许的附件扩展名:

 只允许这些扩展名结尾的附件使用远程附件功能, 每行一个, 不区分大小写, 留空为不限制
 双击输入框可扩大/缩小

禁止的附件扩展名:

 禁止这些扩展名结尾的附件使用远程附件功能, 每行一个, 不区分大小写, 留空为不限制
 双击输入框可扩大/缩小

配置项	说明
启用远程附件	选择是。
启用SSL连接	选择否。

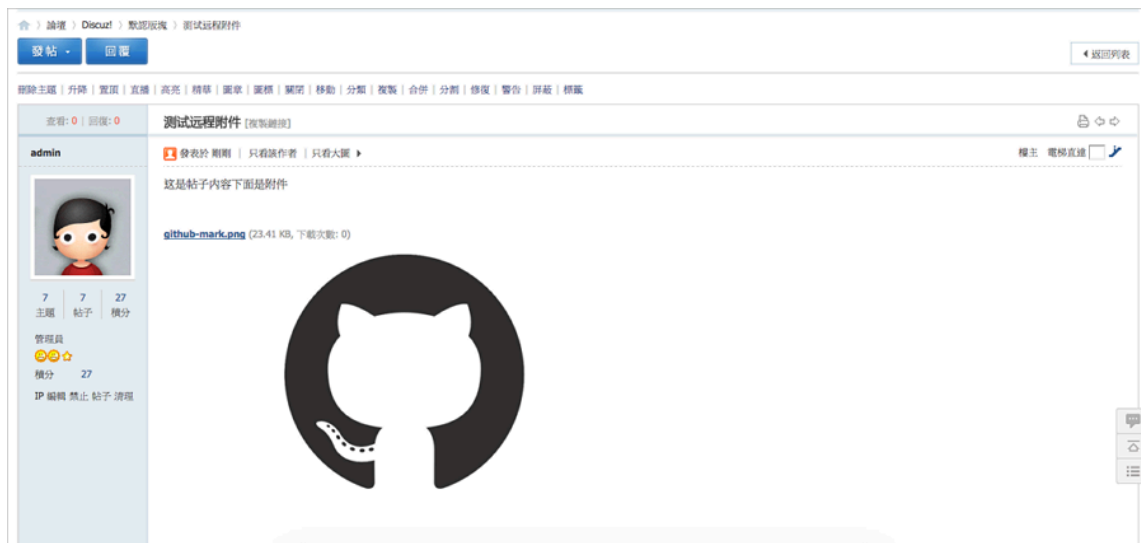
配置项	说明
FTP服务器地址	即运行ossftp工具的地址，通常填写127.0.0.1即可。
FTP服务器端口	默认为2048。
FTP帐号	格式为AccessKeyID/BucketName。注意这里的正斜线 (/) 不是或的意思。
FTP密码	即AccessKeySecret。
被动模式连接	选择是。
远程附件目录	填半角句号 (.) 即可，表示在Bucket的根目录下创建上传目录。
远程访问URL	填写Bucket的外网访问域名，格式为http://BucketName.Endpoint。测试所用Bucket名为test-hz-jh-002，属于杭州地域。所以这里填写的是http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com。关于访问域名的详情请参见 #unique_144 。
超时时间	设置为0，表示服务器默认。

4. 完成设置后，可以单击测试远程附件确认配置是否正常。



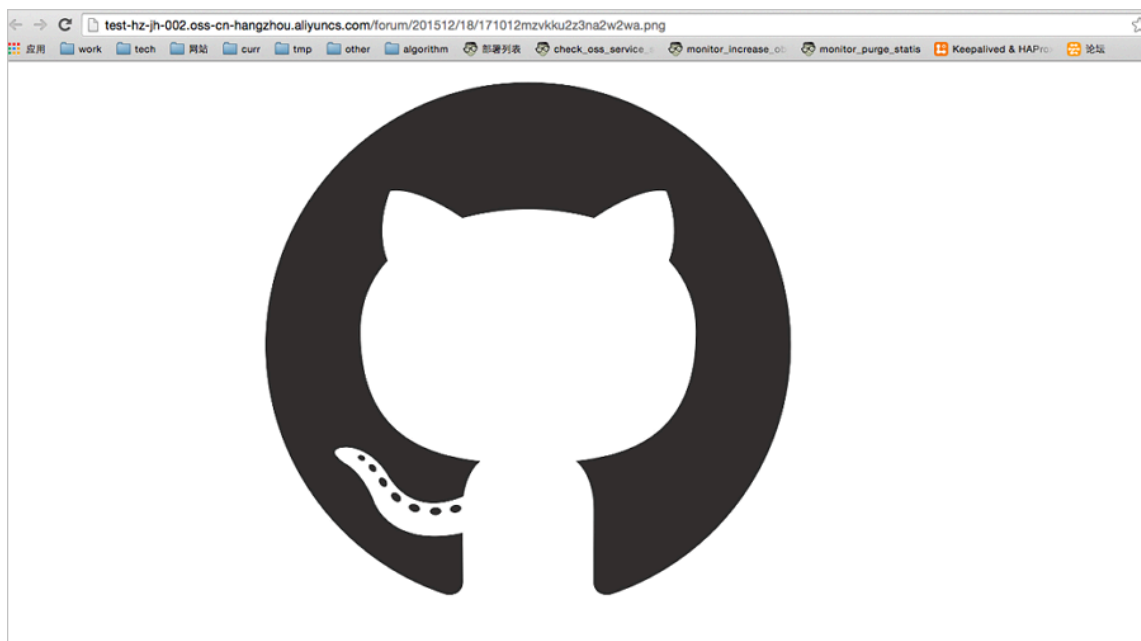
5. 发帖验证配置是否成功。

a) 发帖时上传图片附件。



b) 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，可以判断图片已经上传到OSS的test-hz-jh-002 Bucket。



6.3 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[#unique_142](#)。
 - 创建Bucket的步骤请参见[#unique_143](#)。
- 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。

phpwind

返回旧版

常用

创始人

全局

用户

内容

数据

应用

运营

模式

后台搜索

当前位置 > 全局 > 附件设置 > 附件设置

附件设置

水印设置

图片缩略

FTP设置

全局

站点设置

导航设置

注册设置

积分设置

会员相关

编辑器

附件设置

附件设置

附件优化

附件修复

附件统计

安全优化

SEO设置

消息设置

搜索设置

电子邮件

网上支付

帮助中心

伪静态

FTP设置

使用FTP方式上传

开启

关闭

1

远程附件功能。可将附件上传到远程服务器。尤其针对镜像站点，可将附件传至主站点，便于附件的统一管理

站点附件地址

http://test-hz-jh-002.oss-cn-h

2

源站点附件地址。如: http://www.phpwind.net/attachment

FTP服务器地址

121.40.198.184

3

FTP远程服务器地址

FTP服务器端口

2048

4

FTP服务器端口。默认为21

FTP上传目录

.

5

请确保该目录等同于源站点的attachment目录

FTP帐号

tSxyiUM3NKswPMEp/test-hz

6

FTP帐号

FTP密码

R*****8

7

FTP密码

FTP超时[秒]

10

8

系统将在设定时间内等待服务器的响应

提交

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 #unique_144 。
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/BukcetName。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。

a) 发帖时上传图片附件。



b) 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 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[#unique_142](#)。
- 创建Bucket的步骤请参见[#unique_143](#)。
- 已搭建Wordpress个人网站。

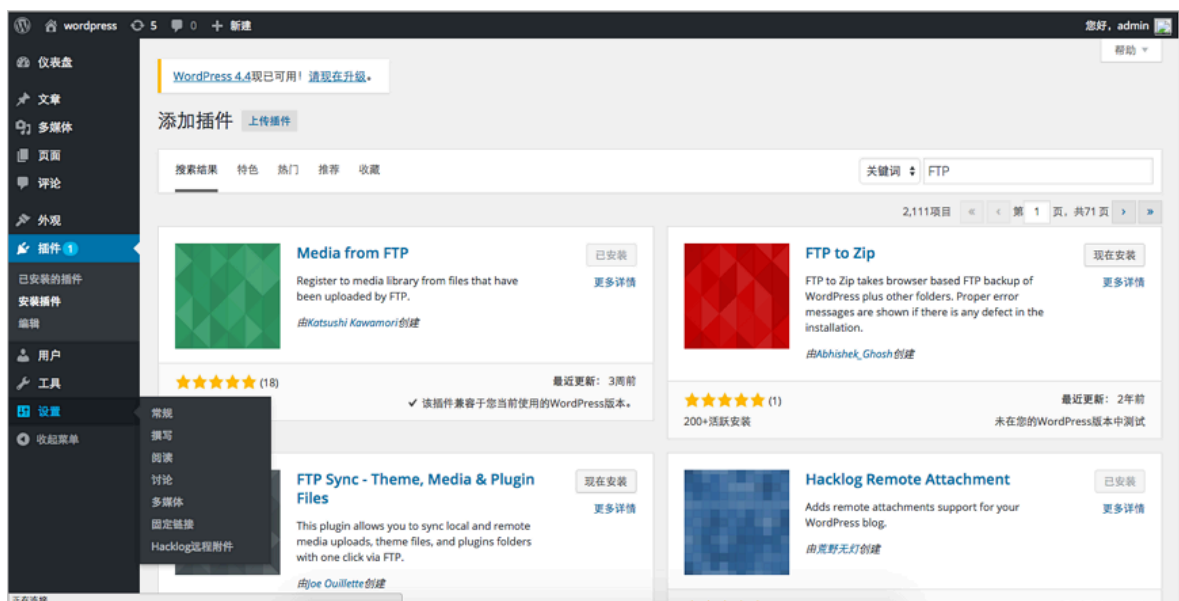
背景信息

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/BukcetName。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#unique_144。
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。

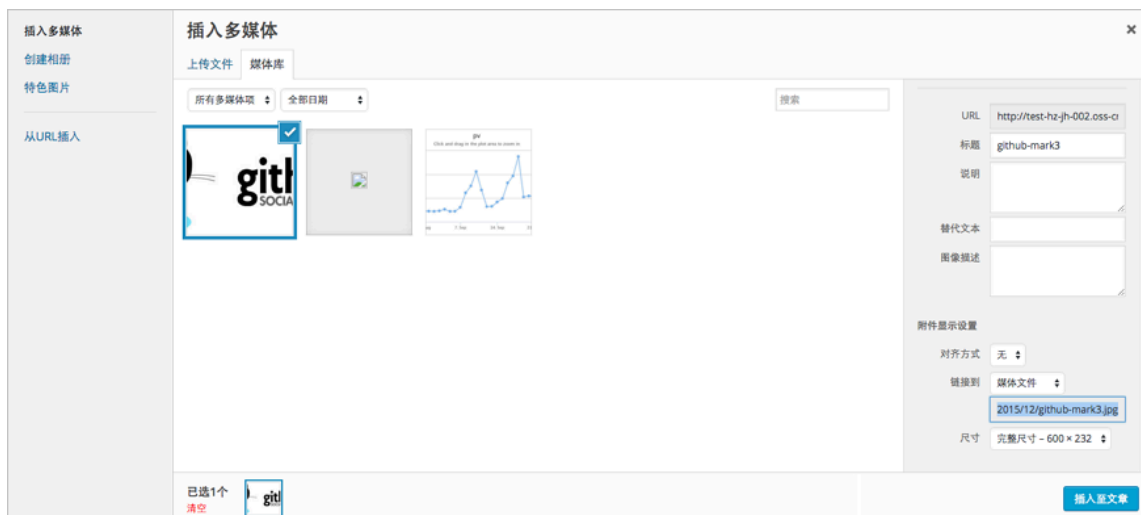


7. 发布新文章验证配置是否成功。

a) 撰写新文章时单击添加媒体来上传附件。



上传附件如下图所示。



b) 单击发布，即可看到刚撰写的文章。



c) 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 如何结合RAM实现文件共享

本文主要介绍如何结合RAM服务，共享用户Bucket中的文件以及文件夹，同时让其他用户拥有该Bucket的只读权限，而Bucket的owner则可以对该bucket进行修改。

按如下操作步骤，即可实现结合RAM服务实现文件共享。

获取账号ID

获取您的账号ID。具体参考下图步骤。



开通RAM

访问控制（Resource Access Management，RAM）是一个稳定可靠的集中式访问控制服务。您可以通过定制策略生成一个共享读的账户，然后使用此账户登录FTP工具并读取您的文件。

RAM的位置请参考下图。



新建授权策略

开通RAM之后，进入RAM控制台，单击左侧的授权管理，按下图步骤操作创建新的授权策略。

填写授权策略时，如下图所示。

创建授权策略

STEP 1: 选择权限策略模板

STEP 2: 编辑权限并挂

• 授权策略名称:

1

read-list-test-hz-john-001

长度为1-128个字符，允许英文字母、数字

备注:

2

read-list-test-hz-john-001

策略内容:

3

```

1 {
2   "Version": "1",
3   "Statement": [
4     {
5       "Action": [
6         "oss:GetObject",
7         "oss:HeadObject",
8       ],
9       "Resource": [
10        "acs:oss:*:[REDACTED]:[REDACTED]:john-001/*"
11      ],
12       "Effect": "Allow"
13     },
14     {
15       "Action": [

```

授权策略格式定义

授权策略常见问题

其中第1步和第2步自定义填写即可，第3步的策略内容填写可参考如下示例。

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "oss:GetObject",
        "oss:HeadObject"
      ],
      "Resource": [
        "acs:oss:*:*****:test-hz-john-001/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "oss:ListObjects",
        "oss:GetBucketAcl",
        "oss:GetBucketLocation"
      ],
      "Resource": [
        "acs:oss:*:*****:test-hz-john-001"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "oss:ListBuckets"
      ],
      "Resource": [
        "acs:oss:*:*****:*"
      ],
      "Effect": "Allow"
    }
  ]
}
```



说明:

将上面的*****替换为您自己的账户ID，test-hz-john-001替换为您自己的Bucket名，然后整体拷贝到策略内容，最后单击新建授权策略即可。

创建用户

以上的授权策略生成了一种只读策略，下面新建一个用户并给予该用户只读策略。新建用户步骤如下：

云

管理控制台

产品与服务 ▾

☰

访问控制RAM

概览

用户管理 1

群组管理

授权策略管理

角色管理

设置

👤

💰

💰

✉

✎

🛡

用户管理

登录名 ▾

请输入登录名

登录名/显示名

test-user-01

test-user-01

创建用户

* 登录名


显示名

备注

邮箱

国家/地区

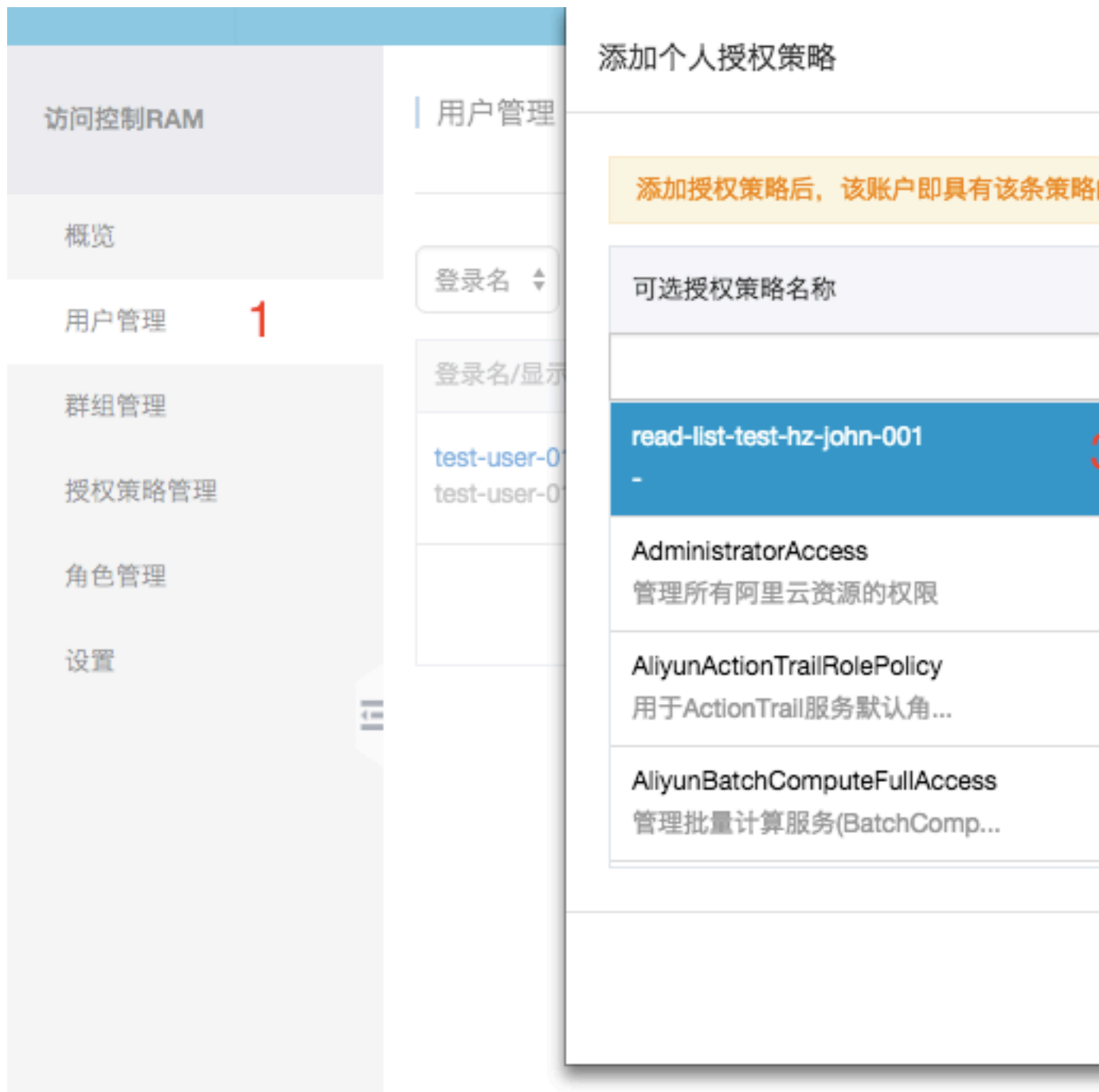
电话

 说明:

注意保存新用户的AccessKey。

给用户授权

将之前创建的策略授权给该用户。



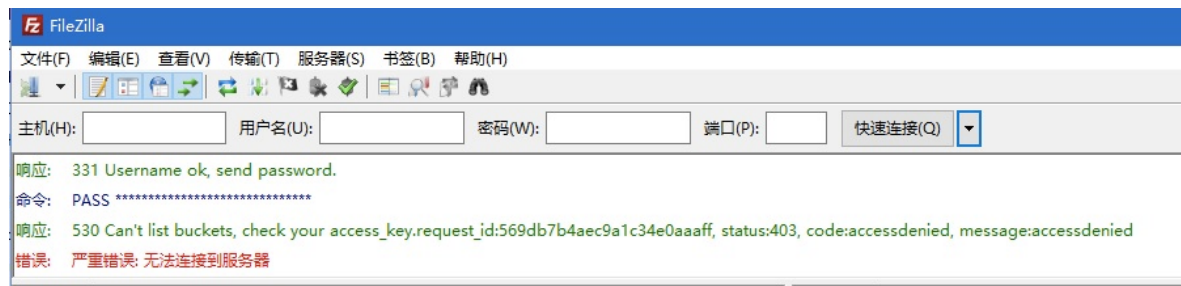
用子账户登录

用子账户的AccessKey和之前授权策略中的Bucket登录，即可下载文件夹或文件，但上传会失败。

6.6 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，无法连接到服务器的



有两种可能：

- 输入的AccessKeyID和AccessKeySecret有误。

解决：请输入正确的信息后再重试。

- 所用的AccessKey信息为RAM子账户的AccessKey，而子账户不具有List Buckets权限。

解决：当使用子账户访问时，请在控制页面中指定Bucket endpoints，即告诉FTP Server某个Bucket应该用哪个Endpoint来访问。同时，子账户也需要一些必须的权限。关于使用RAM访问OSS时的访问控制，请参考文档[访问控制](#)。具体如下：

■ 只读访问

OSS FTP工具需要的权限为 ListObjects、GetObject、HeadObject。关于如何创建一个具有只读访问的RAM子账户，请参考图文教程[如何结合ram实现文件共享](#)。

■ 上传文件

如果允许RAM子账户上传文件，还需要PutObject权限。

■ 删除文件

如果允许RAM子账户删除文件，还需要DeleteObject权限。

- 在Linux下运行FTP Server，之后用FileZilla连接时遇到501错误

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

此类错误通常是因为本地的中文编码有问题。在将要运行start.sh的终端中输入下面的命令，然后再重新启动即可。

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

- 登录成功后List文件超时导致连接断开

原因一般是Bucket根目录下的文件数或文件夹过多。登录ftp后，ftpserver会尝试将Bucket根目录下的所有文件／文件夹List出来，每次可以List出1000个文件／文件夹。如果根目录下有100万以上的文件／文件夹，则会导致1000次以上的http请求，非常容易超时。

- 运行ftpserver的机器由于端口限制导致数据传输不成功

由于ftp协议的控制端口和数据端口不同，当ftpserver工作在被动模式下，每当需要传输数据时，ftpserver会打开1个随机端口，等待客户端来连接。所以当ftpserver所在机器有端口限制时，可能会导致数据无法正常传输。

解决办法是当运行ftpserver.py时，通过指定--passive_ports_start和--passive_ports_end参数来设置本地端口的起止范围，然后将该范围的端口都打开。

- 客户端和ftpserver之间的连接经常断开

各个ftp客户端一般都有超时设置，可以设置为不超时。以FileZilla工具为例，在设置->连接里，可以将超时设置为0。

7 ossfs

7.1 快速安装

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。



说明：

您也可以使用云存储网关服务来访问对象存储OSS，OSS存储资源会以Bucket为基础映射成本地文件夹或者磁盘。

- 云存储网关提供了NFS和SMB（CIFS）两种文件访问协议，从而实现基于OSS的共享文件夹访问。
- 云存储网关还提供了iSCSI协议，将海量的OSS存储空间映射为本地磁盘，并提供高性价比的存储扩容方案。

相比于ossfs，建议您优先使用阿里云云存储网关产品，详情请参见[云存储网关](#)。

使用限制

ossfs使用有以下限制：

- 不支持挂载归档型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`list directory`，因为需要远程访问OSS服务器，所以性能较差。
- 重命名文件/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。例如，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。

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，包括文件读写、目录、链接操作、权限、uid/gid、以及扩展属性（extended attributes）。
- 使用OSS的multipart功能上传大文件。
- 支持MD5校验，保证数据完整性。

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 7.0 (x64)	ossfs_1.80.6_centos7.0_x86_64.rpm
CentOS 6.5 (x64)	ossfs_1.80.6_centos6.5_x86_64.rpm

快速安装

由于低版本的Linux系统内核版本比较低，ossfs进程在运行过程中容易出现掉线或者其他问题。因此建议您将操作系统升级到CentOS 7.0或者Ubuntu 14.04及以上版本。

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系统的安装命令：

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

· CentOS6.5及以上系统版本的安装命令：

```
sudo yum localinstall your_ossfs_package
```



说明：

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，无法直接用yum下载依赖包。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
```

如果需要下载其他依赖包，则将fuse换成对应包的名称。

- CentOS5系统版本的安装命令：

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

3. 配置账号访问信息。

将Bucket名称以及具有此Bucket访问权限的AccessKeyId/AccessKeySecret信息存放在/etc/passwd-ossfs文件中。注意这个文件的权限必须正确设置，建议设为640。

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

4. 将Bucket挂载到指定目录。

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

挂载示例：将杭州地域，名称为bucket-test的Bucket挂载到/tmp/ossfs目录下。

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



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，在这个示例中您可以将OSS endpoint 改为oss-cn-hangzhou-internal.aliyuncs.com，这样可以节省流量费用。OSS的内网域名请参考[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
```



说明：

更多详细信息，请参考[GitHub ossfs](#)。

版本日志

版本日志信息，请参考[GitHub ChangeLog](#)。

7.2 常见问题

本文主要介绍使用ossfs时可能遇到的问题及解决方案。

ossfs适合什么样的程序

ossfs能把OSS Bucket挂载到本地，如果您使用的软件不支持OSS，但您又想让数据能自动同步到OSS，那么ossfs是很好的选择。

ossfs有什么局限性

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io要求很高的应用跑在ossfs挂载的盘上，请慎重考虑。和本地文件系统具体差异：

- 随机或者追加写文件会导致整个文件的重写。
- 元数据操作，例如list directory，性能较差，因为需要远程访问OSS服务器。
- 文件/文件夹的rename操作不是原子的。
- 多个客户端挂载同一个OSS Bucket时，依赖用户自行协调各个客户端的行为。例如避免多个客户端写同一个文件等等。
- 不支持hard link。

ossfs一定要阿里云的机器上使用么

ossfs不限制一定要阿里云的内网才可以使用，外网机器依然可以使用。

ossfs能不能同时挂载多个OSS Bucket

在`passwd-ossfs`文件中写入多个OSS配置信息即可，支持不同帐号的OSS。

使用yum/apt-get安装ossfs，遇到“conflicts with file from package fuse-devel”的错误

您的系统中存在老版本的fuse，请先使用相关的包管理器卸载，再重新安装ossfs。

ossfs工作不正常，如何debug

- 您可以在挂载时加上`-d -o f2`参数，ossfs会把日志写入到系统日志中。CentOS系统在`/var/log/messages`中。
- 您也可以挂载时使用`-f -d -o f2`参数，ossfs会把日志输出到屏幕上。

在mount时遇到“ossfs: unable to access MOUNTPOINT /tmp/ossfs: Transport endpoint is not connected”错误

1. 请先umount对应的目录。
2. 检查您在使用ossfs挂载时，填入的url参数是否正确，是否和Bucket、AccessKeyId、AccessKeySecret匹配。

3. 特别注意：`url`中不包含Bucket的名字。例如：您在OSS控制台中看到Bucket的域名是这样的：`ossfs-test-1.oss-cn-hangzhou.aliyuncs.com`。那么填入的url则是：`http://oss-cn-hangzhou.aliyuncs.com`。

ossfs挂载时提示“`ossfs: unable to access MOUNTPOINT /tmp/odat: No such file or directory`”

这是您未创建该目录导致的，在挂载前需要创建对应目录。

将Bucket挂载到本地后，ls目录时出现“`operation not permitted`”错误

请检查您的Bucket中，是否包含含有不可见字符的OSS 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。使用其他工具对这些Object重命名后，ls就能正确显示目录内容了。

目录下有非常多的文件，为什么ls该目录很慢

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。您可以采用下面两个办法优化：

- 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4MB内存，请根据您的机器内存大小调整为合适的值。
- 使用`ls -f`命令，这个命令会消除与OSS的N次http请求。



说明：

更多详情请参见[issue 13](#)。

ossfs挂载时如何设置权限

如果要允许其他用户访问挂载文件夹，可以在运行ossfs的时候指定`allow_other`参数：

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

使用`allow_other`参数，仍然不能访问文件

`allow_other`是赋予挂载目录其他用户访问的权限，不是里面的文件。如果您要更改文件夹中的文件，请用`chmod`命令。

`allow_other`默认赋予挂载目录777权限，如何将挂载目录的权限设置为770

可以通过`umask`来设置，详情请参见[如何实现770挂载权限](#)。

如何使挂载的文件夹(/tmp/ossfs)属于某个user

- 方法一：如果要使挂载的文件夹(/tmp/ossfs)属于某个user，则需要以user的身份创建挂载文件夹和使用ossfs：

```
sudo -u user mkdir /tmp/ossfs
sudo -u user ossfs bucket-name /tmp/ossfs
```

- 方法二：首先通过cd命令获得指定用户的uid/gid信息。例如获取www用户的uid/gid信息：
id www，然后挂载时指定uid/gid参数：

```
ossfs your_bucket your_mountpoint -ourl=your_url -ouid=your_uid -ogid=your_gid
```

非root用户如何umount ossfs挂载的目录

您可以执行如下命令：

```
fusermount -u your_mountpoint
```

如何开机自动挂载ossfs

1. 请参考[快速安装](#)，将Bucket name、AccessKeyId/AccessKeySecret等信息写入/etc/passwd-ossfs，并将该文件权限修改为640。

```
echo your_bucket_name:your_access_key_id:your_access_key_secret >/etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
```

2. 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。

- Ubuntu14.04和CentOS6.5系统版本通过fstab的方式自动mount：

- a. 在/etc/fstab中加入下面的命令：

```
ossfs#your_bucket_name your_mount_point fuse _netdev,url=your_url,allow_other 0 0
```



说明：

命令中的your_xxx信息需要根据您的Bucket name等信息填入。

- 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，Ubuntu14.04就能自动挂载了。CentOS6.5还需要执行如下命令：

```
chkconfig netfs on
```

- CentOS7.0及以上的系统通过开机自动启动脚本mount：

- 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。

需要以www用户挂载ossfs，如何设置开机自动挂载

参考[如何开机自动挂载ossfs](#)中的配置步骤。其中CentOS7.0及以上的系统的配置步骤需要按照如下方式调整：

1. 修改/etc/init.d/ossfs中的命令为：

```
sudo -u www ossfs your_bucket your_mountpoint -ourl=your_url
```

2. 设置自动启动脚本中允许使用sudo，编辑/etc/sudoers，将其中的Defaults requiretty这行改为#Defaults requiretty（注释掉）。

遇到“fusermount: failed to open current directory: Permission denied”错误如何解决

这是fuse的一个Bug，它要求当前用户对当前目录（非挂载目录）有读权限。解决的办法就是cd到一个有读权限的目录，再运行ossfs命令。

使用ECS挂载ossfs时，如何避免因后台程序扫描文件而产生费用

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（1分钱/1万次）。如果是[updatedb](#)，可以通过修改/etc/updatedb.conf让它跳过。具体做法是：

1. 在PRUNEFS =后面加上fuse.ossfs。
2. 在PRUNEPATHS =后面加上挂载的目录。

使用ECS挂载ossfs时，如何确定是哪个进程扫描了我的目录

1. 首先安装auditd:

```
sudo apt-get install auditd
```

2. 启动auditd:

```
sudo service auditd start
```

3. 设置监视挂载目录:

```
auditctl -w /mnt/ossfs
```

4. 在auditlog中就可以查看是哪些进程访问了这个目录:

```
ausearch -i | grep /mnt/ossfs
```

使用ossfs上传到OSS的文件Content-Type全是” application/octet-stream”

ossfs通过查询/etc/mime.types中的内容来确定文件的Content-Type，请检查这个文件是否存在，如果不存在，则需要添加：

1. 对于Ubuntu可以通过`sudo apt-get install mime-support`来添加。
2. 对于CentOS可以通过`sudo yum install mailcap`来添加。
3. 也可以手动添加，每种格式一行，每行格式为：`application/javascript js`。

如何使用supervisor启动ossfs

1. 安装supervisor，在ubuntu中执行`sudo apt-get install supervisor`。
2. 建立一个目录，编辑ossfs的启动脚本：

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

写入如下数据：

```
# 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 my-bucket my-mount-point -ourl=my-oss-endpoint -f
```

3. 编辑/etc/supervisor/supervisord.conf，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
```

supervisord

5. 确认一切正常:

```
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 # 杀掉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
```

如果出错, 请检查/var/log/supervisor/supervisord.log和/var/log/ossfs.log。

安装ossfs时遇到” fuse: warning: library too old, some operations may not work” 报错

出现的原因是: 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。使用我们提供的rpm包安装ossfs, 无需再安装libfuse。

在CentOS-5.x和CentOS-6.x上我们提供的rpm包里包含了libfuse-2.8.4, 如果在运行的时候环
境中有libfuse-2.8.3, 并且ossfs被链接到了旧版本的fuse上, 就会出现上述warning。

- 如何确认ossfs运行时链接的fuse版本

运行`ldd $(which ossfs) | grep fuse`, 如过结果是” /lib64/libfuse.so.2”, 那么通
过`ls -l /lib64/libfuse*`可以看到fuse的版本。

- 如何让ossfs链接到正确的版本

1. 通过`rpm -ql ossfs | grep fuse`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`
...运行ossfs。

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

因为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/osscli等) 对文件进行了修
改, 那么有可能在ossfs中看到的文件信息没有及时更新。

如果想禁止ossfs的缓存, 那么可以在挂载的时候加上如下参数: `-omax_stat_cache_size=0`。

7.3 ossfs使用问题排查案例

本文介绍在使用 ossfs 时遇到的一些问题案例及解决方案。

ossfs 的报错都会有明显的 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，使用前先将 debug-log 功能开启。

- 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。

如果发现 ossfs 在 ls 目录文件时很慢，可以增加调优参数，通过 `-omax_stat_cache_size=xxx` 参数增大 stat cache 的 size，修改参数之后，第一次 ls 会较慢，但是后续 ls 速度会提高。

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 fil
e 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：这种问题是 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。详情请参考：[fuse](#)。

案例：ossfs 偶尔出现断开的情况

```
Nov 02:58:28 i1zw9exgcy43hdoulltcoez kernel: [30273] 0 30273 33017 827 61 0 0 AliYunDun
Nov 02:58:28 i1zw9exgcy43hdoulltcoez kernel: [16398] 0 16398 1301998 317459 955 0 0 java
Nov 02:58:28 i1zw9exgcy43hdoulltcoez kernel: Out of memory: Kill process 3017 (osfs) score 581 or sacrifice child
Nov 02:58:28 i1zw9exgcy43hdoulltcoez kernel: Killed process 3017 (osfs) total-vms:6846372kB, anon-rss:4787892kB, file-rss:208kB, shmem-rss:0kB
Nov 02:58:45 i1zw9exgcy43hdoulltcoez s3fs[2968]: s3fs.cpp:lst_bucket(2459): xmlReadMemory returns with error.
Nov 02:58:45 i1zw9exgcy43hdoulltcoez s3fs[2968]: s3fs.cpp:s3fs_readdir(2382): list_bucket returns error(-1).
Nov 02:58:45 i1zw9exgcy43hdoulltcoez s3fs[2968]: s3fs.cpp:lst_bucket(2459): xmlReadMemory returns with error.
Nov 02:58:45 i1zw9exgcy43hdoulltcoez s3fs[2968]: s3fs.cpp:s3fs_readdir(2382): list_bucket returns error(-1).
Nov 02:59:01 i1zw9exgcy43hdoulltcoez systemd: Started Session 46140 of user root.
```


解决方案：确认正确的配置信息并修改。

案例：cp 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，用户是对一个 141GB 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@e653a8832c9eb4d8d91 ~]# ossfs --d /mnt/ossfs --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.
```

问题分析：ossfs 上传大文件时，是通过分片来上传的。分片大小默认为 10MB，分片最大数量为 1000 个。

ossfs 在上传文件时会写一些临时缓存文件到 `/tmp` 目录下，在写这些文件之前需要先判断 `/tmp` 目录所在的磁盘可用空间是否小于用户上传的文件总量，若判断磁盘可用空间小于用户上传文件总量，就会出现本地磁盘可用空间不足的报错。以下场景会导致磁盘可用空间不足的报错：

- 场景一：磁盘可用空间本身小于用户上传文件总量。例如，磁盘可用空间是200GB，上传的文件是300GB。
- 场景二：分片大小和上传线程数量的参数设置错误。例如：磁盘可用空间是300GB，需上传的文件是100GB。因操作错误，`multipart_size` 被设置成了100GB，上传线程数量是5。此时`ossfs`判断上传的文件就是 $100\text{GB} \times 5 = 500\text{GB}$ ，超过磁盘安全空间了。

解决方案:

- 场景一：增大磁盘可用空间。
- 场景二：分片大小正常单位是 MB，最大数量是1000，不要将分片大小设置过大。

案例：OSS 挂载到本地，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_hidden000000027000000006]
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] curl.cpp:prepare_url(4030): URL is http://oss-cn-hangzhou-internal.aliyuncs.com/pgback/,fuse_hidden000000027000000006
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_hidden000000027000000006
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] curl.cpp:PutHeadRequest(2369): creating [path=/,fuse_hidden000000027000000006]
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/rewkjt1amz]
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21840][INF] fdcache.cpp:RowFlush(1302): [tpath=][path=/tmp/7KAcr][fd=6]
Dec 22 17:21:22 rerpdb1601 s3fs[21838]: [tid-21839][INF] s3fs.cpp:s3fs_release(2185): [path=/tmp/7KAcr][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。
- 登录的 Accesskey 无该存储空间的操作权限。

解决方案：

- 归档类文件问题：将文件解冻后访问。
- 权限问题：为登录的 Accesskey 对应的账号配置正确的权限。

8 osscmd (已下线)

8.1 快速安装

osscmd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:

osscmd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

osscmd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

- osscmd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。
- osscmd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osscmd所在目录直接执行`python osscmd + 操作`即可。比如上传一个文件到Bucket：

```
python osscmd put myfile.txt oss://mybucket
```



说明:

osscmd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 osscmd`。

如果需要详细的参数列表说明输入：`python osscmd help`。

8.2 使用示例

本文主要以示例的形式介绍osscmd的使用方法。



注意:

osscmd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。

安装配置osscmd

在Linux或者Windows上下载SDK安装包后，解压缩后就可以使用 osscmd了。

使用时直接调用Python osscmd即可获取相应的说明。每种命令有两种执行模式。以查询用户所创建的Bucket为例，执行的是gs命令。

- 方法1：不指定ID和KEY，osscmd从默认文件中读取ID和KEY。

```
$ python osscmd 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 osscmd gs
2013-07-19 08:11 test-oss-sample
Bucket Number is: 1
```

- 方法2：直接在命令中指定ID和KEY，osscmd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 osscmd gs --id=your_id --key=your_key --host=your_endpoint
2013-07-19 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 osscmd 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 osscmd 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 osscmd 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 osscmd getallbucket
```

如果没有成功请检查osscmd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 osscmd 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 osscmd put local_existed_file oss://mybucketname/test_objec
t
```



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

```
0.150(s) elapsed
```

- 读取lifecycle

```
python osscmd getlifecycle oss://mybucket
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>1125</ID>
    <Prefix>log_backup</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>2</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
0.027(s) elapsed
```

- 删除lifecycle

```
python osscmd deletelifecycle oss://mybucket
0.139(s) elapsed
```

- 读取lifecyle

```
python osscmd getlifecycle oss://mybucket
Error Headers:
[('content-length', '288'), ('server', 'AliyunOSS'), ('connection',
'close'), ('x-oss-request-id', '54C74FEE5D7F6B24E5042630'), ('date',
'Tue, 27 Jan 2015 08:44:30 GMT'), ('content-type', 'application/xml')]
Error Body:
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <BucketName>mybucket</BucketName>
  <Code>NoSuchLifecycle</Code>
  <Message>No Row found in Lifecycle Table.</Message>
  <RequestId>54C74FEE5D7F6B24E5042630</RequestId>
  <HostId>mybucket.oss-maque-hz-a.alibaba.net</HostId>
</Error>
Error Status:
404
getlifecycle Failed!
```

防盗链设置

- 允许空referer访问

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

- 获取设置的referer

```
$osscmd 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test的请求

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

- 获取设置的referer

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

- 不允许空referer, 只允许referer为test和test1的请求

```
$osscli putreferer oss://test --allow_empty_referer=false --referer='www.test.com,www.test1.com'
```

- 获取设置的referer

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

使用logging

- 设置logging

```
$osscli putlogging oss://mybucket oss://myloggingbucket/mb
```

- 获取设置的logging

```
$osscli getlogging oss://mybucket
```

8.3 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 osscmd config --id=your_id --key=your_key`
- `python osscmd 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getallbucket是同样的效果。

使用示范:

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

deletebucket(db)

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 osscmd deletebucket oss://mybucket`
- `python osscmd 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 osscmd 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 osscmd 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 osscmd 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 osscmd 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:

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

获取该Bucket lifecycle规则。

使用示范:

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

deletelifecycle

命令说明:

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

删除该bucket下所有的lifecycle规则。

使用示范:

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

putreferer

命令说明:

```
osscmd 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，比如“www.test1.com,www.test2.com”，以“,”作为分隔。详细的配置规则参考[产品文档](#)。

使用示范:

```
python osscmd putreferer oss://mybucket --allow_empty_referer=true
--referer="www.test1.com,www.test2.com"
```

getreferer

命令说明:

```
osscmd getreferer oss://bucket
```

获取该Bucket下防盗链设置规则。

使用示范:

- python osscmd getreferer oss://mybucket

putlogging

命令说明:

```
osscmd 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 osscmd getlogging oss://mybucket
```

getlogging

命令说明:

```
osscmd getlogging oss://bucket
```

获取该bucket的logging设置规则。

使用示范:

```
python osscmd getlogging oss://mybucket
```

8.4 有关Object命令

本文主要介绍与对象（Object）相关的命令。



注意:

osscmd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Object。

使用示范:

- `python osscmd ls oss://mybucket/folder1/folder2`
- `python osscmd ls oss://mybucket/folder1/folder2 marker1`
- `python osscmd ls oss://mybucket/folder1/folder2 marker1 /`
- `python osscmd ls oss://mybucket/`
- `python osscmd list oss://mybucket/ "" "" 100`

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 osscmd ls oss://mybucket/folder1/folder2 --delimiter=/`
- `python osscmd ls oss://mybucket/folder1/folder2 --marker=a`
- `python osscmd ls oss://mybucket/folder1/folder2 --maxkeys=10`

mkdir

命令说明:

```
mkdir oss://bucket/dirname
```

创建一个文件夹。

使用示范：

```
python osscmd 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 osscmd listallobject oss://mybucket`
- `python osscmd 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 osscmd deleteallobject oss://mybucket`
- `python osscmd 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 osscmd downloadallobject oss://mybucket /tmp/folder`
- `python osscmd downloadallobject oss://mybucket /tmp/folder`

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

downloadtodir

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

uploadfromdir

命令说明:

```
uploadfromdir localdir oss://bucket/[prefix] --check_point=check_poin  
t_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, osscmd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 osscmd uploadfromdir /mytemp/folder oss://mybucket`
- `python osscmd uploadfromdir /mytemp/folder oss://mybucket --check_point_file=/tmp/mytemp_record.txt`
- `python osscmd uploadfromdir C:\Documents and Settings\User\My Documents\Downloads oss://mybucket --check_point_file=C:\cp.txt`

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

upload**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 osscmd 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 osscmd 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 osscmd multiget oss://mybucket/myobject /tmp/localfile`
- `python osscmd 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 osscmd 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 osscmd meta oss://mybucket/myobject
```

copy

命令说明:

```
copy oss://source_bucket/source_object oss://target_bucket/target_obj  
ect
```

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

signurl(sign)

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提供给他人访问。

使用示范：

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

8.5 有关Multipart命令

本文主要介绍与碎片（part）相关的命令。



注意：

osscmd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。

使用示范：

```
python osscmd 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。

使用示范：

```
python osscmd 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。

使用示范：

```
python osscmd 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。

使用示范：

```
python osscmd 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。

使用示范：

```
python osscmd 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。

使用示范：

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

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。

使用示范：

- `python osscmd multiupload /tmp/localfile.txt oss://mybucket/object`
`--upload_id=`
`D9D278DB6F8845E9AFE797DD235DC576`
- `python osscmd multiup_load /tmp/localfile.txt oss://mybucket/object`
`--thread_num=5`
- `python osscmd mp /tmp/localfile.txt oss://mybucket/object --max_part_n`
`um=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。

使用示范:

```
python osscmd 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。

9 通过数据集成导入数据

数据集成（Data Integration）是阿里集团对外提供的可跨异构数据存储系统的、可靠、安全、低成本、可弹性扩展的数据同步平台，为400对数据源提供不同网络环境下的全量/增量数据进出通道。

详细的数据源类型列表请参见：[支持数据源类型](#)。用户可以通过[数据集成（Data Integration）](#)对云产品OSS进行数据的导入和导出。

数据导入和导出均有以下两种实现方式：

- 向导模式：可视化界面配置同步任务，一共涉及到五步，选择来源，选择目标，字段映射，通道控制，预览保存。在每个不同的数据源之间，这几步的界面可能有不同的内容，向导模式可以转换成脚本模式。
- 脚本模式：进入脚本界面你可以选择相应的模板，此模板包含了同步任务的主要参数，然后补全剩余的参数也能创建同步任务。但是脚本模式不能转化成向导模式。

本文主要介绍如何将MaxCompute中的数据导入到OSS中，将OSS中的数据导出到MaxCompute中操作步骤与导入类似，因此本文将不再赘述数据如何导出。

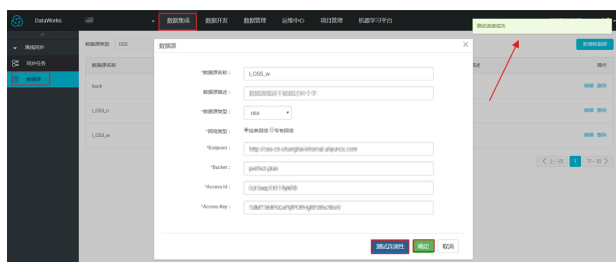


说明：

- 只有项目管理员角色才能够新建数据源，其他角色的成员仅能查看数据源。
- 如您想用子账号创建数据集成任务，需赋予子账号相应的权限。具体请参考：[开通阿里云主账号、设置子账号](#)。

操作步骤

1. 以项目管理员身份进入[数加管理控制台](#)，单击下项目列表下对应项目操作栏中的进入工作区。
2. 进入顶部菜单栏中的数据集成页面，单击左侧导航栏中的数据源。
3. 单击右上角的新增数据源，如下图所示：



4. 在新增数据源对话框中填写相关配置项，针对 OSS 数据源配置项的具体说明如下：

- 数据源名称：由英文字母、数字、下划线组成且需以字符或下划线开头，长度不超过 60 个字符。
- 数据源描述：对数据源进行简单描述，不得超过 80 个字符。
- 数据源类型：当前选择的数据源类型 OSS。
- 经典网络：IP 地址由阿里云统一分配，配置简便，使用方便，适合对操作易用性要求比较高，需要快速使用 ECS 的用户。
- 专有网络：隔离的私有网络，用户可以自定义网络拓扑和 IP 地址，支持通过专线连接，适合对网络管理比较熟悉的用户。
- Endpoint：OSS Endpoint 信息，格式为：`http://region.aliyuncs.com`，OSS 服务的 Endpoint 和 region 有关，访问不同的 region 时，需要填写不同的域名。
- Bucket：相应的 OSS Bucket 信息，存储空间是用于存储对象的容器，可以创建一个或者多个存储空间，然后向每个存储空间中添加一个或多个文件。此处填写的存储空间将在数据同步任务里找到相应的文件，其他没有添加的 Bucket 则不能搜索其中的文件。
- Access Id/Aceess Key：获取方法请参考[获取Access Key](#)和[AccessKeyId](#)。

5. 完成上述信息项的配置后，单击测试连通性。测试通过单击确定。



说明：

其他的数据源的配置请参见：[数据源配置](#)。

6. 新建同步任务，单击数据集成下的同步任务，并选择向导模式。

7. 数据源选择 MaxCompute (原 ODPS) 数据，源头表选择表 mytest，数据预览默认是收起的。

单击下一步，如下图所示：



8. 导入目标选择 OSS 为数据源，并选择相应的 Object，配置项说明如下所示：

- 数据源：选择内容跟填写的数据源名称保持一致。
- object 前缀：填写 object 路径时不要包含 bucket 名，直接取 bucket 后面的内容，如test1是bucket名为test的一级目录，这里object直接填test1。
- 列分隔符：读取的字段分隔符，默认值为“，”。
- 编码格式：文件的编码配置，默认值为 utf-8。
- null值：文本文件中无法使用标准字符串定义 null（空指针），数据同步系统提供 nullFormat 定义哪些字符串可以表示为 null。

具体填写页面如下图所示：



9. 单击下一步选择字段的映射关系。需对字段映射关系进行配置，左侧源头表字段和右侧目标表字段为一一对应的关系，如下图所示。

新建运行停止转换脚本提交

选择来源选择目标

3 字段映射

4 通道控制

5 预览保存

您要配置来源表与目标表映射关系，通过连线将待同步的字段左右相连，也可以通过同行映射批量完成映射。[数据同步文档](#)

源头表字段	类型		目标表序列
id	BIGINT	→	第0列
name	STRING	→	第1列
sex	STRING	→	第2列
age	STRING	→	第3列
添加一行 +			

同行映射

上一步下一步



说明：

单击添加一行：

- 可以输入常量，输入的值需要使用英文单引号包括，如 'abc' 、 '123' 等。

- 可以配合调度参数使用，如’ `${bdp.system.bizdate}`’ 等。
- 可以输入你要同步的分区列，如分区列有pt等。
- 如果您输入的值无法解析，则类型显示为’ - ’。
- 不支持配置MaxCompute函数。

10.单击下一步进入通道控制，配置作业速率上限和脏数据检查规则，如下图所示：

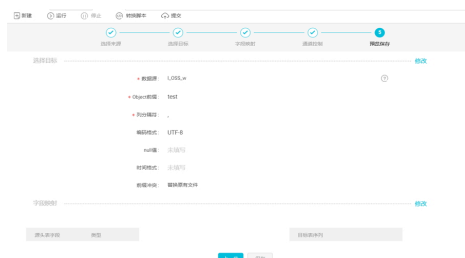


- 作业速率上限：是指数据同步作业可能达到的最高速率，其最终实际速率受网络环境、数据库配置等的影响。
- 作业并发数：作业速率上限=作业并发数*单并发的传输速率。

当作业速率上限已选定的情况下，应该如何选择作业并发数：

- 如果你的数据源是线上的业务库，建议您不要将并发数设置过大，以防对线上库造成影响。
- 如果您对数据同步速率特别在意，建议您选择最大作业速率上限和较大的作业并发数。

11.单击下一步预览保存任务。上下滚动鼠标可查看任务配置，如若无误，单击保存，如下图所示：



12.同步任务保存后，单击运行任务会立刻运行同步任务。单击右边的提交，将同步任务提交到调度系统中，调度系统会按照配置属性从第二天开始自动定时执行，任务运行结束即可将MaxCompute中的数据导入到OSS中。运行后的提示界面如下如所示。

```
Writer: drds
      postSql=[[]]
      shared=[false]
      *password=[*****]
      column=[["id"]]
      description=[]
      gmtCreate=[2017-06-05 11:20:10]
      type=[drds]
datasourceNetwork=[classic]
datasourceType=[drds]
datasourceBackUp=[l_Drds_w
      jdbcUrl=[jdbc:mysql://drds5c08708617e9public.drds.aliyuncs.com:3306/cdpctest331]
      name=[l_Drds_w]
      tenantId=[177437243534241]
      subType=[]
      id=[56975]
      projectId=[40978]
      table=[contact_infos]
      preSql=[[]]
      status=[1]
      username=[cdptest331]
2017-07-07 14:30:08 : State: 2(WAIT) | Total: 0R 0B | Speed: 0R/s 0B/s | Error: 0R 0B | Stage: 0.0%
2017-07-07 14:30:18 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Error: 0R 0B | Stage: 0.0%
2017-07-07 14:30:28 : State: 0(SUCCESS) | Total: 3R 24B | Speed: 0R/s 2B/s | Error: 3R 24B | Stage: 100.0%
2017-07-07 14:30:28 : CDP Job[39171410] completed successfully.
2017-07-07 14:30:28 : ---
CDP Submit at      : 2017-07-07 14:30:08
CDP Start at      : 2017-07-07 14:30:11
CDP Finish at     : 2017-07-07 14:30:23
2017-07-07 14:30:28 : Use "cdp job -log 39171410 [-p basecommon_group_177437243534241_cdp_dev]" for more detail.
```

相关调度的配置请参考[调度配置介绍](#)。

您在新建同步任务的时候也可以选择使用脚本模式配置同步任务来进行数据导入：

```
{
  "configuration": {
    "reader": {
      "plugin": "odps",
      "parameter": {}
    },
    "writer": {
      "plugin": "oss",
      "parameter": {
        "fieldDelimiterOrigin": ",",
        "datasource": "l_OSS_w", //数据源名，建议数据源都先添加数据源后再配置同步任务，此配置项填写的内容必须与添加的数据源名称保持一致
        "column": [ //列名
          "0",
          "1",
          "2",
          "3"
        ],
        "writeMode": "truncate", //写入模式
        "encoding": "UTF-8", //编码格式
        "fieldDelimiter": ",", //分隔符
        "object": "test" //object路径
      }
    },
    "setting": {
      "speed": {
        "concurrent": 7, //并发的数目

```

```
        "mbps": 9//一个并发的速率上线是9MB/S
    }
  },
  "type": "job",
  "version": "1.0"
}
```



说明:

使用向导模式将 OSS 中的数据导出到 MaxCompute 的步骤与上述将 MaxCompute 中的数据导入到OSS类似。