

Alibaba Cloud

对象存储

常用工具

档案版本：20191015

目錄

1 OSS常用工具匯總.....	1
2 ossutil.....	3
2.1 下載和安裝.....	3
2.2 查看選項.....	8
2.3 有關bucket的命令.....	10
2.4 有關object的命令.....	12
2.5 有關multipart的命令.....	15
3 ossimport.....	17
3.1 說明及配置.....	17
3.2 單機部署.....	30
3.3 分布式部署.....	34
3.4 資料移轉.....	37
3.5 常見問題.....	42
4 RAM策略編輯器.....	50
5 ossftp.....	52
5.1 如何快速安裝OSS FTP.....	52
5.2 Discuz如何儲存遠程附件到OSS.....	55
5.3 Phpwind如何儲存遠程附件到oss.....	57
5.4 Wordpress如何儲存遠程附件到oss.....	58
5.5 如何結合RAM實現檔案分享權限設定.....	59
6 ossfs.....	62
6.1 快速安裝.....	62
6.2 FAQ.....	64
7 osscmd.....	70
7.1 快速安裝.....	70
7.2 使用樣本.....	71
7.3 有關Bucket命令.....	76
7.4 有關Object命令.....	80
7.5 有關Multipart命令.....	86

1 OSS常用工具匯總

OSS除了控制台還有以下常用工具，可以幫助您更高效的使用OSS。

工具	簡介	備忘
ossbrowser	圖形化的Object管理工具。支援Windows、Linux、Mac平台。	官方工具。提供類似Windows資源管理員的功能。用戶可以方便的瀏覽檔案、上傳下載檔案、支援斷點續傳等。
ossutil	命令列管理工具。提供方便、簡潔、豐富的Object管理命令。	官方工具，支援Windows, Linux, Mac平台，不依賴於任何第三方組件，下載後即用不需要安裝。
osscmd	命令列管理工具。提供完備的Bucket、object管理命令。	官方工具。基於Python2.5 - 2.7版本，支援多平台。將逐步被ossutil替代，除非需要ossutil不具備的Bucket管理功能外，強烈推薦使用ossutil。
ossfs	掛載bucket到本地檔案系統，能夠通過本地檔案系統操作OSS上的對象，實現數據的訪問和共用。	官方工具。支援Linux平台。
ossftp	FTP工具，使用FTP協議來管理OSS的object，可以使用FileZilla、WinSCP、FlashFXP等FTP客戶端操作OSS。OSSFTP本質是FTP Server，接收FTP請求，將對檔案、檔案夾的操作映射為對OSS的操作。	官方工具。基於Python2.7及以上，支援Windows、Linux、Mac平台。
ossimport	資料同步工具。可以將本地或第三方雲端儲存體服務上的檔案同步到OSS上。	官方工具。依賴JRE7及以上。支援Windows、Linux平台。
可視化圖片服務工具	可以直觀的看到OSS圖片服務處理的結果，圖片處理的調試不可或缺的工具。	第三方工具。網頁版。支援瀏覽器Chrome、Firefox、Safari。

工具	簡介	備忘
可視化簽名工具	可視化簽名工具。可以調試簽名遇到的問題。當您實現OSS的簽名中遇到問題，請跟該工具的簽名對比，錯誤立現。	第三方工具。網頁版。支援瀏覽器Chrome、Firefox、Safari。
<i>RAM Policy Editor</i>	OSS授權策略自動化生產工具。當您需要生成自己的授權策略時，強烈推薦使用該工具。	官方工具。網頁版。支援瀏覽器Chrome、Firefox、Safari。
<i>oss-emulator</i>	輕量級的OSS服務模擬器，用於基於OSS應用的調試、效能測試等。	官方工具。支援Windows、Linux、Mac。

2 ossutil

2.1 下載和安裝

ossutil工具旨在為用戶提供一個方便的，以命令列方式管理OSS數據的途徑。目前的版本未提供完整的Bucket管理功能和Multipart管理功能，相關功能會在後續版本中開發。現在如果有使用上述功能的需要，可以先使用`osscmd`命令列工具。

工具下載

- 目前的版本
 - 目前的版本：1.4.1
- 運行環境
 - Windows/Linux/Mac
 - 支援架構
 - x86 (32bit, 64bit)
- binary下載
 - [Linux x86 32bit] [ossutil32](#)
 - [Linux x86 64bit] [ossutil64](#)
 - [Windows x86 32bit] [ossutil32.zip](#)
 - [Windows x86 64bit] [ossutil64.zip](#)
 - [mac x86 64bit] [ossutilmac64](#)

- 安裝使用

根據您的作業系統選擇相應的binary或者壓縮包下載後，運行相應的binary（如果binary為不可執行檔，請給binary增加可執行許可權：`chmod 755 ossutil`），即：

linux系統下：

```
./ossutil
```

windows系統下有兩種方法（以64位系統為例）：

1) 解壓壓縮包，雙擊運行其中的bat檔案，再鍵入：`ossutil64.exe`

2) 解壓壓縮包，cmd進入壓縮包中binary所在的目錄，鍵入：`ossutil64.exe`

mac系統下：

```
./ossutilmac64
```

快速使用

- 設定ossutil的語言

在使用ossutil的命令時，可以使用-L選項設定語言，可選範圍為CH/EN，即：中文或英文。

大小寫不敏感。預設語言為CH（中文），如果設定成CH（中文），需要確保您的系統為utf-8編碼，否則可能會顯示亂碼。

如：

```
./ossutil help ls顯示ls預設語言的幫助
```

```
./ossutil help ls -L ch顯示ls的中文幫助
```

```
./ossutil help ls -L en顯示ls的英文幫助
```

```
./ossutil config -L ch運行ossutil config的互動式配置命令，其中的提示言為中文。
```

```
./ossutil config -L en運行ossutil config的互動式配置命令，其中的提示言為英文。
```



说明：

ossutil輸出的錯誤預設都為英文，不會受上述選項影響。

- 獲取命令列表

```
./ossutil或./ossutil help
```

```
$./ossutil
用法: ossutil [command] [args...] [options...]
請使用ossutil help command來顯示command命令的幫助
Commands:
    mb                cloud_url [options]
```

```

ls      建立Bucket          [cloud_url] [options]
rm      列舉Buckets或者Objects
        cloud_url [options]
stat    刪除Bucket或Objects
        cloud_url [options]
set-acl 顯示bucket或者object的描述資訊
        cloud_url [acl] [options]
set-meta 設定bucket或者objects的acl
        cloud_url [meta] [options]
cp      設定已上傳的objects的元資訊
        src_url dest_url [options]
restore 上傳, 下載或拷貝Objects
        cloud_url [options]
create-symlink 恢復冷凍狀態的Objects為可讀狀態
        cloud_url target_url [options]
read-symlink 建立符號連結
        cloud_url [options]
Additional Commands:
help    讀取符號連結檔案的描述資訊
        [command]
config  獲取命令的幫助文檔
        [options]
hash    建立設定檔用以儲存配置項
        file_url [options]
update  計算本地檔案的crc64或md5
        [options]
        更新ossutil

```

```

$./ossutil -L en
Usage: ossutil [command] [args...] [options...]
Please use 'ossutil help command' to show help of command
Commands:
mb      cloud_url [options]
        Make Bucket
ls      [cloud_url] [options]
        List Buckets or Objects
rm      cloud_url [options]
        Remove Bucket or Objects
stat    cloud_url [options]
        Display meta information of bucket or objects
set-acl cloud_url [acl] [options]
        Set acl on bucket or objects
set-meta cloud_url [meta] [options]
        set metadata on already uploaded objects
cp      src_url dest_url [options]
        Upload, Download or Copy Objects
restore cloud_url [options]
        Restore Frozen State Object to Read Ready Status
create-symlink cloud_url target_url [options]
        Create symlink of object
read-symlink cloud_url [options]
        Display meta information of symlink object
Additional Commands:
help    [command]
        Get help about commands
config  [options]
        Create configuration file to store credentials
hash    file_url [options]
        Get crc64 or md5 of local file
update  [options]

```

Update ossutil

- 查看某命令的幫助文檔

`./ossutil help cmd` 強烈建議在使用某命令前先使用 `help` 來查閱幫助文檔。

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

SYNOPSIS

建立設定檔用以儲存配置項

SYNTAX

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

DETAIL DESCRIPTION

該命令建立設定檔，將使用者佈建的配置項資訊儲存進該設定檔，配置項用

以訪問OSS時提供訪問資訊（某命令是否需要配置項，參見其是否支援

`--config-file` 選項，具體可見該命令的幫助）。

設定檔路徑可由用戶指定，預設為 `/home/admin/.ossutilconfig`。如果配置檔案存在，假設其為 `a`，`ossutil` 會將檔案 `a` 另存新檔：`a.bak`，然後重新建立檔案 `a` 並寫入配置，此時，如果 `a.bak` 存在，其會被檔案 `a` 覆蓋。

注意：

(1) 如果指定的設定檔路徑非預設路徑，在使用命令時請將 `--config-file` 選項設定為你配置時指定的設定檔路徑（如果不指定 `--config-file` 選項，則運行命令時預設會讀取 `/home/admin/.ossutilconfig`）。

(2) 某些配置可在使用命令時通過選項進行設定，如 `--endpoint`，`--access-key-id`，

`id`，

等選項（具體請見每個命令的幫助），如果使用命令時指定了這些選項，並且同時設定檔中也配置了這些資訊，則優先順序為：選項 > 設定檔。

(3) 如果使用命令時指定了 `--endpoint`、`--access-key-id`、`--access-key-secret`

`secret`

或 `--sts-token` 選項，則 `ossutil` 不強求設定檔一定要存在。

用法：

該命令有兩種用法，互動式1) 和非互動式2)，推薦用法為互動式，因為互動式用法擁有更好的安全性。

1) `ossutil config [-c file]`

該用法提供一種互動方法來配置資訊，`ossutil` 互動式地詢問用戶如下資訊：

(1) `config file`

設定檔路徑，如果用戶鍵入回車，`ossutil` 會使用預設的設定檔：`/home/admin/.ossutilconfig`。

如果用戶自己指定了設定檔，在使用命令時需要將 `--config-file` 選項設定為使用者佈建的設定檔路徑。哪些命令支援 `--config-file` 選項可由查看每個命令的幫助。

(2) `language`

當首次配置（設定檔不存在）時，`ossutil` 會向用戶詢問語言設定，可選值為中文或者英文（CH/EN），如果鍵入回車，`ossutil` 將根據用戶輸入的 `--language` 選項配置，如果此時用戶也未輸入 `--language` 選項，將配置成預設語言中文。

如果設定檔已存在，`ossutil` 會綜合用戶輸入的 `language` 選項和設定檔中的語言資訊，配置該項，而不會詢問。

`ossutil` 在運行時會從設定檔中讀取該 `language` 選項，如果該選項不存在或者非法，將採用預設語言：CH。

注意：該配置項在此次 `config` 成功結束後才會生效，在執行 `config` 命令過程中語言顯示不會受用戶的選擇影響。

(3) `endpoint`，`accessKeyID`，`accessKeySecret`

回車代表著跳過相應配置項的設定。注意：`endpoint` 應該為一個二級域名（SLD），例如：`oss.aliyuncs.com`。

以上選項一般為必選項。

(4) `stsToken`

如果用戶需要使用臨時 `token` 來訪問 `oss`，用戶需要填入該項，否則請輸入回車跳過該項配置。

(5) `outputDir`

該選項配置輸出檔案所在目錄的路徑。互動式模式時不提供該選項的配置，但設定檔中該項配置起效。

outputDir的預設目錄為：目前的目錄下的：ossutil_output，ossutil會在運行過程中將輸出檔案都生成到該檔案夾下。輸出檔案目前包含：在cp命令中批量操作出錯時，記錄每個檔案操作的錯誤資訊的report檔案。關於outputDir和report檔案的更多資訊請參見cp命令的幫助。注意：outputDir如果不存在，ossutil在產生輸出檔案時會自動建立該目錄，

如果outputDir存在且並非目錄，將會報錯。下述互動式Bucket-Endpoint和Bucket-Cname配置被取消，但設定檔中該兩項配置仍然起效。

(6) Bucket-Endpoint
Bucket-Endpoint對每個指定的bucket單獨配置endpoint，此配置會優先於設定檔中關於預設endpoint的配置。在該版本中，ossutil取消了互動式配置中，關於Bucket-Endpoint配對配置，但設定檔中該項配置仍然起效，所以如果用戶想對每個bucket單獨指定endpoint，仍然可以在設定檔中進行配置。注意：此處的endpoint應該為一個次層網域(SLD)，例如：oss.aliyuncs.com。如果配置了Bucket-Endpoint選項，當對某bucket進行操作時，ossutil會在該選項中尋找該bucket對應的endpoint，如果找到，該endpoint會覆蓋基本配置中endpoint。但是運行命令時如果指定了--endpoint選項，--endpoint選項為最高優先順序。

(7) Bucket-Cname
Bucket-Cname為每個指定的bucket單獨配置CNAME網域名稱（CDN加速網域名稱），此配置會優先於設定檔中Bucket-Endpoint及endpoint的配置。在該版本中，ossutil取消了互動式配置中，關於Bucket-Cname配對的配置，但設定檔中該項配置仍然起效，所以如果用戶想對每個bucket單獨指定CNAME網域名稱，仍然可以在設定檔中進行配置。如果配置了Bucket-Cname選項，當對某bucket進行操作時，ossutil會在該選項中尋找該bucket對應的CNAME網域名稱，如果找到，則找到的CNAME網域名稱會覆蓋Bucket-Endpoint選項和基本配置中的endpoint。運行命令時如果指定了--endpoint選項，--endpoint選項為最高優先順序。

優先順序：--endpoint > Bucket-Cname > Bucket-Endpoint > 預設endpoint

2) ossutil config options
如果用戶使用命令時輸入了除--language和--config-file之外的任何選項，則該命令進入非互動式模式。所有的配置項應當使用選項指定。

設定檔格式：

```
[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
...
```

SAMPLE
ossutil config

```
ossutil config -e oss-cn-hangzhou.aliyuncs.com -c ~/.myconfig
```

OPTIONS

- c, --config-file
ossutil 的設定檔路徑，ossutil 啟動時從設定檔讀取配置，在 config 命令中，ossutil 將配置寫入該檔案。
- e, --endpoint
ossutil 工具的基本 endpoint 配置（該選項值會覆蓋設定檔中的相應設定），注意其必須為一個次層網域。
- i, --access-key-id
訪問 oss 使用的 AccessKeyID（該選項值會覆蓋設定檔中的相應設定）。
- k, --access-key-secret
訪問 oss 使用的 AccessKeySecret（該選項值會覆蓋設定檔中的相應設定）。
- t, --sts-token
訪問 oss 使用的 STSToken（該選項值會覆蓋設定檔中的相應設定），非必須設定項。
- output-dir=ossutil_output
指定輸出檔案所在的目錄，輸出檔案目前包含：cp 命令批量拷貝檔案出錯時所產生的 report 檔案（關於 report 檔案更多資訊，請參考 cp 命令幫助）。預設值為：目前的目錄下的 ossutil_output 目錄。
- L CH, --language=CH
設定 ossutil 工具的語言，預設值：CH，取值範圍：CH/EN，若設定成"CH"，請確保您的系統編碼為 UTF-8。

• 配置 ossutil

在使用訪問 oss 的命令時，需要先配置訪問 AK，關於 AK 的更多資訊見：[RAM](#) 和 [STS](#) 介紹。

配置 ossutil 由兩種方式：[互動式](#) 和 [非互動式](#)。

關於配置命令的更多幫助，請使用 `ossutil help config` 查看。

- 互動式配置 ossutil

```
./ossutil config
```

```
$./ossutil config -L ch
```

該命令建立將一個設定檔，在其中儲存配置資訊。

請輸入設定檔路徑（預設為：`/home/admin/.ossutilconfig`，回車將使用預設路徑。如果使用者佈建為其它路徑，在使用命令時需要將 `--config-file` 選項設定為該路徑）：

- 非互動式配置 ossutil

```
./ossutil config -e oss.aliyuncs.com -i your_id -k your_key
```

2.2 查看選項

通過 `-h` 選項來查看所有 ossutil 支援的所有 option。

```
$./ossutil -h
```

Usage of ossutil:

Options:

```
-s --short-format
```

顯示精簡格式，如果未指定該選項，預設顯示長格式。

```
--snapshot-path=
```

該選項用於在某些場景下加速增量上傳批量檔案（目

前，下載和拷貝不支援該選項）。在 cp 上傳檔案時使用該選項，ossutil 在指定的目錄下組建檔案記錄檔案上傳的快照資訊，在下次指定該選項上傳時，ossutil 會讀取指定目錄下的快照資訊進行增量上傳。用戶指定的 snapshot 目錄必須為本地檔案系統上的可寫目錄，若該目錄不存在，ossutil 會建立該檔案用於記錄快照資訊，如果該目錄已存在，ossutil 會讀取裡面的快照資訊，根據快照資訊進行增量上傳（只上傳上次未成功上傳的檔案和本地進行

過修改的檔案)，並更新快照資訊。注意：因為該選項通過在本地記錄成功上傳的檔案的本地lastModifiedTime，從而在下次上傳時通過比較lastModifiedTime來決定是否跳過相同檔案的上傳，所以在使用該選項時，請確保兩次上傳期間沒有其他用戶更改了oss上的對應object。當不滿足該場景時，如果想要增量上傳批量檔案，請使用--update選項。另外，ossutil不會主動刪除snapshot-path下的快照資訊，為了避免快照資訊過多，當用戶確定快照資訊無用時，請用戶自行清理snapshot-path。

-j --jobs= 多檔案操作時的並發任務數，預設值：5，取值範圍：1-10000

-v --version 顯示ossutil的版本 (1.0.0.Beta2) 並退出。

--output-dir= 指定輸出檔案所在的目錄，輸出檔案目前包含：cp命令批量拷貝檔案出錯時所產生的report檔案（關於report檔案更多資訊，請參考cp命令幫助）。預設值為：目前的目錄下的ossutil_output目錄。

--parallel= 單檔案內部操作的並發任務數，取值範圍：1-10000，預設將由ossutil根據操作類型和檔案大小自行決定。

-L --language= 設定ossutil工具的語言，預設值：CH，取值範圍：CH/EN

-t --sts-token= 訪問oss使用的STSToken（該選項值會覆蓋設定檔中的相應設定），非必須設定項。

-m --multipart 指定操作的對象為bucket中未完成的Multipart事件，而非預設情況下的object。

-b --bucket bucket 對bucket進行操作，該選項用於確認操作作用於

--delete 刪除操作

-e --endpoint= ossutil工具的基本endpoint配置（該選項值會覆蓋設定檔中的相應設定），注意其必須為一個次層網域。

-k --access-key-secret= 訪問oss使用的AccessKeySecret（該選項值會覆蓋設定檔中的相應設定）。

--bigfile-threshold= 開啟大檔案斷點續傳的檔案大小閾值，預設值：100M，取值範圍：0B-9223372036854775807B

--retry-times= 當錯誤發生時的重試次數，預設值：3，取值範圍：1-500

-a --all-type 指定操作的對象為bucket中的object和未完成的Multipart事件。

-r --recursive 遞迴進行操作。對於支援該選項的命令，當指定該選項時，命令會對bucket下所有合格objects進行操作，否則只對url中指定的單個object進行操作。

-f --force 強制操作，不進行詢問提示。

-u --update 更新操作

-c --config-file= ossutil工具的設定檔路徑，ossutil啟動時從設定檔讀取配置，在config命令中，ossutil將配置寫入該檔案。

-i --access-key-id= 訪問oss使用的AccessKeyID（該選項值會覆蓋設定檔中的相應設定）。

--acl= acl資訊的配置。

-d --directory 返回目前的目錄下的檔案和子目錄，而非遞迴顯示所有子目錄下的所有object。

--checkpoint-dir= checkpoint目錄的路徑(預設值為:.ossutil_checkpoint)，斷點續傳時，操作失敗ossutil會自動建立該目錄，並在該目錄下記錄checkpoint資訊，操作成功會刪除該目錄。如果指定了該選項，請確保所指定的目錄可以被刪除。

--type= 計算的類型，預設值：crc64，取值範圍：crc64/md5

-h --help Show usage message

ossutil的每個命令支援以上option中的部分option，若要查看某個命令command支援哪些option，請使用ossutil help command來查看。

2.3 有關bucket的命令

Bucket的相關命令

ossutil提供了建立、刪除、列舉Bucket、以及為Bucket設定acl的功能，關於Bucket更多的管理功能暫時不支援，如需要請使用`osscmd`。

在使用這些命令前，請先使用`config`命令配置訪問AK。

• 建立Bucket

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

其中acl如果不指定，則預設為private許可權，如果成功建立，ossutil會列印消耗時間並退出，否則會輸出錯誤資訊。可以通過`--storage-class`選項指定儲存方式。

關於建立Bucket的幫助資訊，請使用`ossutil help mb`命令查看。

```
./ossutil mb oss://test  
0.220478(s) elapsed
```

• 刪除Bucket

關於刪除Bucket的幫助資訊，請使用`ossutil help rm`命令查看。注意：

- 刪除bucket必須設定**-b**選項；
- 被刪除的bucket可能被其他用戶重新建立，從而不再屬於您；
- bucket中的數據一旦被刪除則無法恢復。

(1) 如果您的Bucket中沒有數據

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

```
./ossutil rm oss://test -b  
Do you really mean to remove the Bucket: test(y or N)? y  
0.220478(s) elapsed
```

(2) 如果您的Bucket中有object或multipart等數據，需要先刪除所有數據再刪除Bucket，可以使用以下命令來一併刪除所有數據和您的Bucket

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

關於刪除Bucket的幫助資訊，請使用`ossutil help rm`命令查看。

- 列舉Buckets

```
./ossutil ls或 ./ossutil ls oss://
```

可以使用-s選項來顯示精簡格式，更多幫助見: `ossutil help ls`

```

$./ossutil ls
CreationTime                               Region      StorageClass
BucketName
2016-10-21 16:18:37 +0800 CST               oss-cn-hangzhou      Archive
oss://go-sdk-test-bucket-xyz-for-object
2016-12-01 15:06:21 +0800 CST               oss-cn-hangzhou      Standard
oss://ossutil-test
2016-07-18 17:54:49 +0800 CST               oss-cn-hangzhou      Standard
oss://ossutilconfig
2016-07-20 10:36:24 +0800 CST               oss-cn-hangzhou      IA
oss://ossutilupdate
2016-11-14 13:08:36 +0800 CST               oss-cn-hangzhou      IA
oss://yyyyy
2016-08-25 09:06:10 +0800 CST               oss-cn-hangzhou      Archive
oss://ztzt
2016-11-21 21:18:39 +0800 CST               oss-cn-hangzhou      Archive
oss://ztztzt
Bucket Number is: 7
0.252174(s) elapsed

```

- 列舉Bucket中的檔案

ossutil可以列舉Bucket中的Object和UploadID，預設情況下顯示Object，使用-m選項來顯示UploadID，使用-a選項同時顯示Object和UploadID。

- 列舉Object

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

```

$./ossutil ls oss://ossutil-test
LastModifiedTime                               Size(B)  StorageClass  ETAG
ObjectName
2016-12-01 15:06:37 +0800 CST                 10363812  Standard
61DE142E5AFF9A6748707D4A77BFBCFB             oss://ossutil-test/a1
2016-12-01 15:06:42 +0800 CST                 10363812  Standard
61DE142E5AFF9A6748707D4A77BFBCFB             oss://ossutil-test/a2
2016-12-01 15:06:45 +0800 CST                 10363812  Standard
61DE142E5AFF9A6748707D4A77BFBCFB             oss://ossutil-test/a3
Object Number is: 3
0.007379(s) elapsed

```

- 列舉Object和Multipart

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

```

$ 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

```

可以使用-s選項顯示精簡模式。可以使用-d選項顯示首層目錄下的內容。

```

$ ossutil ls oss://bucket1 -d
oss://bucket1/obj1
oss://bucket1/sample.txt
oss://bucket1/dir1/
Object and Directory Number is: 3
UploadID      ObjectName
15754AF7980C4DFB8193F190837520BB      oss://bucket1/obj1
2A1F9B4A95E341BD9285CC42BB950EE0      oss://bucket1/obj1
3998971ACAF94AD9AC48EAC1988BE863      oss://bucket1/obj2
A20157A7B2FEC4670626DAE0F4C0073C      oss://bucket1/tobj
UploadId Number is: 4
0.119884(s) elapsed

```

- 為Bucket設定acl

建立Bucket時，Bucket預設的acl為private，可以通過set-acl命令來修改Bucket的acl。在設定Bucket的acl許可權時，需要設定-b選項。

將bucket1設定為private許可權：

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

關於設定acl的更多資訊請使用help set set-acl來查看。

2.4 有關object的命令

ossutil提供了上傳/下載/拷貝檔案、設定object的acl、設定object的meta、查看object的meta資訊等功能。

使用這些命令前請使用config命令配置訪問AK。

- 上傳/下載/拷貝檔案

強烈建議在使用cp命令前使用ossutil help cp先查看幫助。

可以使用cp命令進行上傳/下載/拷貝檔案，使用-r選項來拷貝檔案夾，對大檔案預設使用分區上傳並可進行斷點續傳（開啟分區上傳的大檔案閾值可用--bigfile-threshold選項來設定）。

使用-f選項來預設強制上傳，當目標端存在同名檔案時，不詢問，直接覆蓋。

當批量上傳/下載/拷貝檔案時，如果某個檔案出錯，ossutil預設會將錯誤資訊記錄在report檔案，並跳過該檔案，繼續其他檔案的操作（當錯誤為Bucket不存在、accessKeyID/accessKeySecret錯誤造成的許可權驗證非法等錯誤時，不再繼續其他檔案拷貝）。更多資訊請見ossutil help cp。

ossutil支援特定場景下的增量上傳策略：--update和--snapshot-path選項，請參見ossutil help cp。

ossutil從1.0.0.Beta1版本開始，上傳檔案預設開啟crc64。

- 上傳單個檔案：

```
./ossutil cp a oss://ossutil-test
Succeed: Total num: 1, size: 230. OK num: 1(upload 1 files).
0.699795(s) elapsed
```

- 上傳檔案夾：

```
./ossutil cp -r dir oss://ossutil-test
Succeed: Total num: 35, size: 464,606. OK num: 35(upload 34 files
, 1 directories).
0.896320(s) elapsed
```

- 上傳/下載/拷貝檔案的性能調優

在cp命令中，通過jobs項和-parallel項控制並發數。-jobs項控制多個檔案上傳/下載/拷貝時，檔案間啟動的並發數。-parallel制分區上傳/下載/拷貝一個大檔案時，每一個大檔案啟動的並發數。

預設情況下，ossutil會根據檔案大小來計算parallel個數（該選項對於小檔案不起作用，進行分區上傳/下載/拷貝的大檔案檔案閾值可由--bigfile-threshold選項來控制），當進行批量大檔案的上傳/下載/拷貝時，實際的並發數為jobs個數乘以parallel個數。該兩個選項可由用戶調整，當ossutil自行設定的預設並發達不到用戶的性能需求時，用戶可以自行調整該兩個選項來升降性能。



说明：

- 如果並發數調得太大，由於線程間資源切換及搶奪等，ossutil上傳/下載/拷貝性能可能會下降，所以請根據實際的機器情況調整這兩個選項的數值，如果要進行壓測，可以一開始將兩個數值調低，慢慢調大尋找最優值。
- 如果--jobs選項和--parallel選項值太大，在機器資源有限的情況下，可能會因為網路傳輸太慢，產生EOF錯誤，這個時候請適當降低--jobs選項和--parallel選項值。

• 設定object的acl

ossutil使用set-acl命令設定object的acl，使用-r選項可以批量設定object的acl。

更多資訊見：[ossutil help set-acl](#)

```
./ossutil set-acl oss://dest/a private
0.074507(s) elapsed
```

批量設定acl：

```
./ossutil set-acl oss://dest/a private -r
Do you really mean to recursively set acl on objects of oss://dest/a
(y or N)? y
Succeed: Total 3 objects. Setted acl on 3 objects.
0.963934(s) elapsed
```

• 設定object的meta

ossutil使用set-meta命令設定object的meta資訊，使用-r選項可以批量設定object的meta

。

更多資訊見：[ossutil help set-meta](#)

```
./ossutil set-meta oss://dest/a x-oss-object-acl:private -u
```



说明：

您可以在上傳檔案的同時設定object的meta資訊，詳情請參見[ossutil的過濾參數include/exclude](#)

。

• 查看object描述資訊 (meta)

ossutil使用stat命令查看object的描述 (meta) 資訊。

更多資訊見：[ossutil help stat](#)

```
./ossutil stat oss://dest/a
ACL : default
Accept-Ranges : bytes
Content-Length : 230
Content-Md5 : +5vbQC/MSQK0xXSiyKBZog==
Content-Type : application/octet-stream
Etag : FB9BDB402FCC4902B4C574A2C8A059A2
Last-Modified : 2017-01-13 15:14:22 +0800 CST
Owner : aliyun
```

```
X-Oss-Hash-Crc64ecma      : 12488808046134286088
X-Oss-Object-Type         : Normal
0.125417(s) elapsed
```

- 恢復冷凍狀態的object為可讀狀態

ossutil使用restore命令恢復冷凍狀態的object為可讀狀態。可以使用-r選項批量恢復冷凍狀態的objects為可讀狀態。

更多資訊見：[ossutil help restore](#)

```
./ossutil restore oss://utiltest/a
0.037729(s) elapsed
```

- 建立符號連結

ossutil使用create-symlink建立符號連結。

更多資訊見：[ossutil help create-symlink](#)

```
./ossutil create-symlink oss://utiltest/b a
0.037729(s) elapsed
```

- 讀取符號連結檔案的描述資訊

ossutil使用read-symlink讀取符號連結檔案的描述資訊。

更多資訊見：[ossutil help read-symlink](#)

```
./ossutil read-symlink oss://utiltest/b
Etag      : D7257B62AA6A26D66686391037B7D61A
Last-Modified : 2017-04-26 15:34:27 +0800 CST
X-Oss-Symlink-Target : a
0.112494(s) elapsed
```

2.5 有關multipart的命令

Multipart的相關命令

ossutil提供了列舉UploadID、刪除指定object的所有UploadID的功能。

關於Multipart的更多介紹請參見[斷點續傳](#)。



说明:

ossutil在上傳/拷貝檔案時，對於大檔案自動進行分區斷點續傳，而不提供UploadPart命令。

- 列舉UploadID

使用-m選項來列舉指定object下的所有未完成UploadID，使用-a選項來列舉object和UploadID。

```
$ 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

```

· 刪除指定object的所有UploadID

使用-m選項刪除指定object下的所有未完成UploadID，當同時指定-r選項時，會刪除以指定object為首碼的所有object的未完成UploadID。

假設bucket1下有如下檔案：

```

$ 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                  241561      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

```

刪除obj1的2個UploadID：

```

$ ./ossutil rm -m oss://bucket1/obj1
Succeed: Total 2 uploadIds. Removed 2 uploadIds.
1.922915(s) elapsed

```

刪除obj1和obj2的3個UploadID：

```

$ ./ossutil rm -m oss://bucket1/obj1
Succeed: Total 4 uploadIds. Removed 4 uploadIds.
1.922915(s) elapsed

```

同時刪除obj1, obj1和obj2的3個UploadID：

```

$ ./ossutil rm oss://dest1/.a -a -r -f
Do you really mean to remove recursively objects and multipart
uploadIds of oss://dest1/.a(y or N)? y
Succeed: Total 1 objects, 3 uploadIds. Removed 1 objects, 3
uploadIds.

```

3 ossimport

3.1 說明及配置

概述

OssImport工具可以將本地、其它雲端儲存的資料移轉到OSS，它有以下特點：

- 支援的豐富的資料來源，有本地、七牛、百度BOS、AWS S3、Azure Blob、又拍雲、騰訊雲COS、金山KS3、HTTP、OSS等，並可根據需要擴充；
- 支援斷點續傳；
- 支援流量控制；
- 支援遷移指定時間後的檔案、特定首碼的檔案；
- 支援並行數據下載、上傳；
- 支援單機模式和分布式模式，單機模式部署簡單使用方便，分布式模式適合大規模資料移轉。

運行環境

- Java 1.7及以上

部署方式

OssImport有單機模式和分布式模式兩種部署方式。

- 對於小於 30TB 的小規模資料移轉，單機模式即可完成。[下載地址](#)
- 對於大規模的資料移轉，請使用分布式模式。[下載地址](#)

單機

Master、Worker、Tracker、Console運行在一個機器上，系統中有且只有一個Worker。

我們對單機模式的部署和執行進行了封裝優化，單機部署和執行都很簡單。單機模式下

Master、Worker、TaskTracker、Console四個模組統一打包成ossimport2.jar。

單機模式下檔案結構如下：

```
ossimport
├── bin
│   └── ossimport2.jar # 包括Master、Worker、Tracker、Console四個模組的總
jar
├── conf
│   ├── local_job.cfg # 單機Job設定檔
│   └── sys.properties # 系統運行參數設定檔
├── console.bat # Windows命令列，可以分布執行調入任務
└── console.sh # Linux命令列，可以分布執行調入任務
```

```

├── import.bat          # Windows一鍵匯入，執行設定檔為conf/local_job.cfg
配置的資料移轉任務，包括啟動、遷移、校驗、重試
├── import.sh          # Linux一鍵匯入，執行設定檔為conf/local_job.cfg配
置的資料移轉任務，包括啟動、遷移、校驗、重試
├── logs                # 日誌目錄
└── README.md          # 說明文檔，強烈建議使用前仔細閱讀

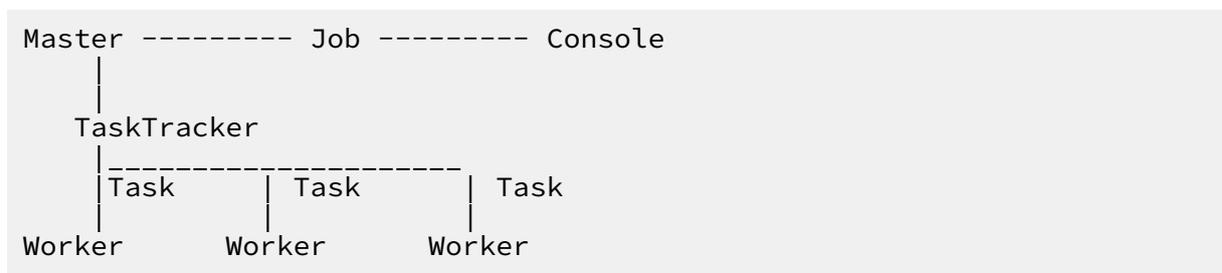
```

提示：

- import.bat/import.sh為一鍵匯入指令碼或直譯式程式，修改完local_job.cfg後可以直接運行；
- console.bat/console.sh為命令列工具，可以用於分布執行命令；
- 指令碼或直譯式程式或命令時請在ossimport目錄下執行，即 *.bat/*.sh 的同級目錄。

分布式

OssImport是基於master-worker的分布式架構，如下圖：



其中：

- Job：用戶通過提交的資料移轉任務，對用戶來說一個任務對應一個設定檔job.cfg。
- Task：Job按照數據大小和檔案個數可以分成多個Task，每個Task 遷移部分檔案。Job切分成Task的最小單位是檔案，同一個檔案不會切分到多個Task中。

OssImport各模組說明，如下表：

角色	說明
Master	負責將Job切分成Task，按照數據大小和檔案個數分解成Task，數據大小/檔案個數可以在sys.properties中配置。Job切分成Task的詳細過程是： 1. Master從本地/其它雲端儲存中遍曆出完整的待遷移的檔案清單。 2. 按照數據大小和檔案個數把完整的檔案清單切分成Task，每個Task負責部分檔案的遷移或校驗。

角色	說明
Worker	<ul style="list-style-type: none"> 負責Task的檔案遷移和數據校驗，從資料來源上拉取指定檔案，並上傳到OSS的指定目錄。遷移的資料來源和OSS的配置在job.cfg或local_job.cfg中指定 Worker資料移轉支援限流、指定Task並發數，在sys.properties中配置。
TaskTracker	簡稱Tracker，負責Task的分發、Task狀態跟蹤。
Console	負責與用戶互動，接受命令顯示結果。支援系統管理命令deploy/start/stop，Job管理命令submit/retry/clean。

分布式模式下可以啟動多個Worker執行遷移數據，Task平均分配到Worker上執行，一個Worker執行多個Task。每一個機器上只能啟動一個Worker。workers配置的第一個Worker上會同時啟動Master、TaskTracker、Console也要在該機器上運行。

分布式模式下檔案結構如下：

```

ossimport
├── bin
│   ├── console.jar # Console模組jar包
│   ├── master.jar # Master模組jar包
│   ├── tracker.jar # Tracker模組jar包
│   └── worker.jar # Worker模組jar包
├── conf
│   ├── job.cfg # Job設定檔範本
│   ├── sys.properties # 系統運行參數設定檔
│   └── workers # Worker列表
├── console.sh # 命令列工具，目前支援只Linux
├── logs # 日誌目錄
└── README.md # 說明文檔，強烈建議使用前仔細閱讀
    
```

提示：

- 分布式命令列工具 console.sh 目前只支援Linux，Windows暫不支援。

設定檔

單機模式下有兩個設定檔sys.properties、local_job.cfg，分布式模式下有三個設定檔sys.properties、local_job.cfg、workers。其中local_job.cfg和job.cfg是完全一樣的，只是名稱不一樣，workers是分布式環境先獨有的。

· **sys.properties**

系統運行參數。

欄位	含義	說明
workingDir	工作目錄	<ul style="list-style-type: none"> - 工具包解壓後的目錄。 - 單機模式下請不要修改此選擇。 - 分布式模式下每個機器的工作目錄必須相同。
workerUser	Worker機器的ssh用戶名	<ul style="list-style-type: none"> - 如果配置了privateKey File，則優先使用privateKeyFile。 - 如果沒有配置privateKeyFile，則使用workerUser/workerPassword。 - 單機模式不需要修改此項。
workerPassword	Worker機器的ssh使用者密碼	單機模式不需要修改此項。
privateKeyFile	private key檔案路徑	<ul style="list-style-type: none"> - 如果已經打通了ssh通道，則可以指定public key檔案路徑，否則為空。 - 如果配置了privateKey File，則優先使用privateKeyFile。 - 如果沒有配置privateKeyFile，則使用workerUser/workerPassword。 - 單機模式不需要修改此項。
sshPort	ssh通信埠	預設22，一般情況無需更改 單機模式不需要修改此項。

欄位	含義	說明
workerTaskThreadNum	Worker執行Task的最大線程數	<ul style="list-style-type: none"> - 該參數與機器的記憶體/網路有關，建議值60。 - 物理機可以適當加大，比如150，如果網路已經打滿，請不要再加大。 - 如果網路較差，請適當降低，比如30，防止請求競爭網路造成大量請求逾時。
workerMaxThroughput(KB/s)	worker資料移轉的資料傳輸量上限	該值能起到限流作用，預設0表示不限流。
dispatcherThreadNum	Tracker的Task分發與狀態確認的線程數	預設值一般夠用，沒有特殊需要請不要修改預設值。
workerAbortWhenUncatchedException	表示遇到未知錯誤時是跳過還是Abort	預設跳過。
workerRecordMd5	在OSS中是否使用元數據x-oss-meta-md5記錄遷移檔案MD5值，預設不記錄。	主要用於用戶使用MD5校驗檔案數據。

· job.cfg

資料移轉任務配置，local_job.cfg和job.cfg的配置項是完全一樣的，只是名稱不一樣。

欄位	含義	說明
jobName	任務名字，字元串。	<ul style="list-style-type: none"> - 任務的唯一標識，命名規則[a-zA-Z0-9_-]{4,128}，支援提交多個名字不同的任務。 - 如果重複提交同名任務會提示任務已存在，清理（clean）同名任務前，無法提交同名任務。
jobType	任務類型，字元串	<p>兩類import或audit，預設import。</p> <ul style="list-style-type: none"> - import，執行資料移轉操作並校驗遷移數據的一致性。 - audit，僅進行數據一致性校驗。

欄位	含義	說明
isIncremental	是否開啟增量移轉模式，布爾類型。	<ul style="list-style-type: none"> - 預設值false。 - 如果設為true，會每間隔 incrementalModeInterval(單位秒)重新掃描一次增量數據，並將增量資料移轉到OSS。
incrementalModeInterval	增量模式下的同步間隔，整形，單位秒。	isIncremental=true 時有效。可配置的最小間隔為900秒，不建議配置成小於3600秒的值，會浪費大量請求，造成額外的系統開銷。
importSince	遷移大於該時間的數據，整形，單位秒。	<ul style="list-style-type: none"> - 該時間為 Unix時間戳記，即自1970年1月1日UTC零點以來的秒數，通過命令 date +%s獲取。 - 預設為0，表示遷移全部數據。

欄位	含義	說明
srcType	同步源類型，字元串，請注意大小寫。	<p>目前支援 local、oss、qiniu、bos、ks3、s3、youpai 等十種類型。</p> <ul style="list-style-type: none"> - local，從本地檔案遷移數據到OSS，該選項只需要填寫srcPrefix，不需要填寫srcAccessKey，srcSecretKey，srcDomain，srcBucket。 - oss，從一個 bucket 遷移到另一個 bucket。 - qiniu，從七牛雲端儲存遷移到OSS。 - bos，從百度的雲端儲存遷移到OSS。 - ks3，從金山雲端儲存遷移到OSS。 - s3，從 AWS S3 遷移到OSS。 - youpai，從又拍雲遷移到OSS。 - http，通過提供的HTTP連結清單遷移數據到OSS。 - cos，從騰訊雲端儲存COS遷移到OSS。 - azure，從Azuer Blob遷移到OSS。
srcAccessKey	源AccessKey，字元串。	<p>如果srcType設定為oss、qiniu、baidu、ks3、s3，填寫資料來源的AccessKey。</p> <ul style="list-style-type: none"> - local、http，該項不需要填寫。 - youpai、azure則填寫用戶名AccountName。

欄位	含義	說明
srcSecretKey	源SecretKey, 字元串。	<p>如果 srcType 設定為 oss、qiniu、baidu、ks3、s3, 填寫資料來源的 SecretKey。</p> <ul style="list-style-type: none">- local、http, 該項不需要填寫。- youpai, 填寫操作員密碼。- azure, 填寫 AccountKey。

欄位	含義	說明
srcDomain	源Endpoint	<p>如果 srcType 設定為 local、http，該配置項不需要填。</p> <ul style="list-style-type: none"> - oss，從控制台獲取的網域名稱，非帶bucket首碼的次層網域，列表請參看網域名稱列表。 - qiniu，從七牛控制台獲取的對應bucket的網域名稱。 - bos，百度BOS網域名稱，如http://bj.bcebos.com 或 http://gz.bcebos.com。 - ks3，金山ks3網域名稱，如http://kss.ksyun.com 或 http://ks3-cn-beijing.ksyun.com 或 http://ks3-us-west-1.ksyun.com。 - S3，AWS S3各 region 的網域名稱請參考S3 Endpoint。 - youpai，又拍雲網域名稱，如自動判斷最優線路 http://v0.api.upyun.com 或電信線路http://v1.api.upyun.com 或聯通網通線路http://v2.api.upyun.com或移動鐵通線路http://v3.api.upyun.com。 - cos，騰訊雲填寫bucket所在的區域，比如華南園區:gz、華北園區:tj、華東園區:sh。 - azure，Azure Blob連接字串中的 EndpointSuffix，如core.chinacloudapi.cn。

欄位	含義	說明
srcBucket	源bucket名字或container名稱。	如果 srcType 設定為local、http，不需要填寫。azure, Azure Blob填寫container名稱，其它填寫bucket名稱。
srcPrefix	源首碼，字元串，預設為空。	如果srcType=local，填寫本地目錄，需要完整路徑，以/進行分割並且以/結尾，如 c:/example/或 /data/example/。srcType 為oss、qiniu、bos、ks3、youpai、s3，則為待同步object的首碼，不包括bucket名稱，如data/to/oss/，同步所有檔案srcPrefix設定為空。
destAccessKey	目的AccessKey，字元串。	OSS的AccessKeyID，請到 阿里雲控制台 查看。
destSecretKey	目的SecretKey，字元串。	OSS的AccessKeySecret，請到 阿里雲控制台 查看。
destDomain	目的endpoint，字元串。	從 阿里雲控制台 獲取，非帶bucket首碼的次層網域，列表請參看網域名稱列表。
destBucket	目的bucket，字元串。	OSS的bucket名稱，不需要以/結尾。

欄位	含義	說明
destPrefix	目標首碼，字元串，預設為空。	<ul style="list-style-type: none"> - 目標首碼，預設為空，直接放在目標bucket下。 - 如果要將資料同步到oss的某個目錄下，請以/結尾，如data/in/oss/。 - 注意oss不支援以/作為檔案的開頭，所以destPrefix請不要配置以/做為開頭。 - 一個本地檔案路徑為srcPrefix+relativePath的檔案，遷移到oss的路徑為destDomain/destBucket/destPrefix+relativePath。 - 一個雲端檔案路徑為srcDomain/srcBucket/srcPrefix+relativePath的檔案，遷移到oss的路徑為destDomain/destBucket/destPrefix+relativePath。
taskObjectCountLimit	每個Task最大的檔案數，整型，預設10000。	該配置項會影響到任務執行的並行度，一般配置為總檔案數/Worker總數/遷移線程數(workerTaskThreadNum)，最大值不要超過50000，如果不知道總檔案數，請使用預設值。
taskObjectSizeLimit	每個Task最大數據量，整型，單位bytes，預設1GB。	該配置項會影響到任務執行的並行度，一般配置為總數據量/Worker總數/遷移線程數(workerTaskThreadNum)，如果不知道總數據量，請使用預設值。
isSkipExistFile	資料移轉時是否跳過已經存在的檔案，布爾類型。	當設定為true時，根據檔案的size和LastModifiedTime判斷是否跳過；為false時，總是覆蓋OSS上已有檔案。預設為值為false。jobType為audit時此項不生效。

欄位	含義	說明
scanThreadCount	並行掃描檔案的線程數，整型，預設1。	該配置項與掃描檔案的效率有關，沒有特殊需求請不要修改。
maxMultiThreadScanDepth	最大允許並行掃描目錄的深度，整型，預設1。	<ul style="list-style-type: none"> - 預設值1表示在頂級目錄間並行掃描。 - 沒有特殊需求不要修改，隨意配置過大值會導致任務無法正常運行。
appId	騰訊雲COS的appId，整型。	srcType=cos時有效。
httpListFilePath	HTTP列表檔案的絕對路徑，字元串。	<ul style="list-style-type: none"> - srcType=http時有效，源為HTTP連結地址時，需要提供內容為HTTP連結地址檔案的絕對路徑，如c:/example/http.list。 - 該檔案中的HTTP連結需要劃分成兩列，以空格分開，分別代表首碼和上傳到OSS後的相對路徑。例如c:/example/http.list檔案內容如：<code>http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/aa/ bb.jpg</code> <code>http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/cc/dd.jpg</code>兩行，遷移到OSS的檔案名分別是 <code>destPrefix + bb.jpg</code>和 <code>destPrefix + cc/dd.jpg</code>。

• workers

workers是分布式模式先獨有的，每個IP一行。如：

```
192.168.1.6
192.168.1.7
```

192.168.1.8

提示：

- 上述配置情況下，第一行的192.168.1.6一定是 master ；即192.168.1.6 上會啟動 Master、Worker、TaskTracker、Console也需要在該機上執行。
- 多個Worker機器的用戶名、登入方式、工作目錄請確保相同。

設定檔樣本

下表中是分布式部署下的資料移轉任務設定檔，單機的設定檔名是local_job.cfg，配置項與分布式部署時沒有區別。

遷移類型	設定檔	說明
從本地遷移到OSS	<i>job.cfg</i>	srcPrefix 是以 / 結尾的絕對路徑，如 D:/work/oss/data/, /home/user/work/oss/data/
從七牛雲端儲存遷移到OSS	<i>job.cfg</i>	srcPrefix 和 destPrefix 可以配置為空；如果不為空，請以 / 結尾，如 destPrefix=docs/
從百度bos遷移到OSS	<i>job.cfg</i>	srcPrefix 和 destPrefix 可以配置為空；如果不為空，請以 / 結尾，如 destPrefix=docs/
從AWS S3遷移到OSS	<i>job.cfg</i>	S3的網域名稱列表
從又拍雲端儲存遷移到OSS	<i>job.cfg</i>	srcAccessKey/srcSecretKey填操作員帳號及密碼
從騰訊cos遷移到OSS	<i>job.cfg</i>	srcDomain請按照V4版本填寫，如srcDomain=sh；srcPrefix可以為空，當不為空時候，請以 / 開頭和結尾，如 srcPrefix=/docs/
從Azure blob遷移到OSS	<i>job.cfg</i>	srcAccessKey/srcSecretKey填儲存儲存帳號及密鑰；srcDomain填連接字串中的 EndpointSuffix，如 core.chinacloudapi.cn

遷移類型	設定檔	說明
從OSS遷移到OSS	job.cfg	適用於不同區域之間、不同儲存類型之間、不同首碼之間的資料移轉；推薦在ECS上部署，並使用帶internal的網域名稱，可以節省流量費用
使用Express Connect遷移數據到OSS	job.cfg	適用於所有資料來源，如果您有高速遷移需求，請提交工單或聯繫OSS技術服務人員

3.2 單機部署

下載

單機部署支援Linux、Windows。

單機版本下載地址[ossimport-2.3.2.zip](#)，下載到本地後，使用工具或命令unzip，解壓後的檔案結構如下：

```

ossimport
├── bin
│   └── ossimport2.jar # 包括Master、Worker、TaskTracker、Console四個模
組的總jar
├── conf
│   ├── local_job.cfg # Job設定檔
│   └── sys.properties # 系統運行參數設定檔
├── console.bat # Windows命令列，可以分布執行調入任務
├── console.sh # Linux命令列，可以分布執行調入任務
├── import.bat # Windows一鍵匯入，執行設定檔為conf/local_job.cfg
├── import.sh # Linux一鍵匯入，執行設定檔為conf/local_job.cfg
├── logs # 日誌目錄
└── README.md # 說明文檔，強烈建議使用前仔細閱讀

```

配置

單機版本有兩個設定檔conf/sys.properties、conf/local_job.cfg。注意不要修改以下內容：

- conf/sys.properties中的配置項workingDir、workerUserName、workerPassword、privateKeyFile
- conf/local_job.cfg的名稱和位置，配置項jobName

其它配置項請按照實際需求配置。



说明:

請在提交任務前確認`sys.properties`和`local_job.cfg`中的參數，任務提交後參數無法再修改。

運行

單機模式下，資料移轉任務有以下兩種執行方式：

- 一鍵匯入：是對所有步驟的封裝，按照指令碼或直譯式程式提示執行即可完成資料移轉。
- 分步執行：執行啟動服務、提交任務、重試失敗子任務等步驟。



说明:

對於初級用戶強烈建議使用一鍵匯入。

- 一鍵匯入

1. 執行一鍵匯入，Window系統下在`cmd.exe`中執行`import.bat`，Linux終端中執行`bash import.sh`。
2. 如果之前執行過程式，會提示有是否從上次的斷點處繼續執行，或者重新執行同步任務。對新的資料移轉任務，或者修改了同步的源端/目的端，請選擇重新執行。
3. Windows下任務開始後，會開啟一個新的`cmd`視窗執行同步任務並顯示日誌，舊視窗會每隔10秒打一次任務狀態，資料移轉期間不要關閉兩個視窗；Linux下服務在後台執行。
4. 當Job完成後，如果發現有任務失敗了，會提示是否重試，輸入`y`重試，輸入`n`則跳過退出。
5. 如果上傳失敗，請開啟檔案`master/jobs/local_test/failed_tasks/<tasktaskid>/audit.log`查看，確定失敗原因。

- 分步執行



说明:

沒有特殊需要，請使用 一鍵匯入方式遷移數據。

1. 清除同名任務。

如果以前運行過同名任務，需要重新執行任務，請先清除同名任務。如果沒有運行過，或需要重試失敗任務，不要執行清除命令。Windows下在cmd.exe中執行console.bat clean，Linux下在終端執行bash console.sh clean。

2. 提交資料移轉任務。

OssImport不能提交同名任務，如果有請先清除。提交任務的設定檔為conf/local_job.cfg，預設任務名稱為local_test。提交任務的命令，Windows下在cmd.exe中執行console.bat submit，Linux下在終端執行bash console.sh submit。

3. 啟動服務。

Windows下在cmd.exe中執行console.bat start，Linux下在終端執行bash console.sh start。

4. 查看任務狀態。

Windows下在cmd.exe中執行console.bat stat，Linux下在終端執行bash console.sh stat。

5. 失敗Task重試。

由於網路或其它原因，子任務可能失敗。失敗重試只重試失敗的Task，不會重試成功的Task。Windows下在cmd.exe中執行console.bat retry，Linux下在終端執行bash console.sh retry。

6. 停止服務。

Windows下在關閉視窗 %JAVA_HOME%/bin/java.exe 即可，Linux下在終端執行bash console.sh stop。

· 常見失敗原因

- 上傳過程中來源目錄的檔案發生了修改，log/audit.log裡會提示SIZE_NOT_MATCH相關字樣的錯誤，這種情況下老的檔案已經上傳成功，新的修改沒有上傳到OSS；
- 源檔案在上傳過程中被刪除，導致下載的時候失敗；
- 源檔案名不符合OSS命名規範（不能以/開頭，不能為空），導致上傳到OSS失敗；
- 下載資料來源檔案失敗；
- 程式異常退出，任務狀態為 Abort，這種情況請聯繫我們。

· 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设定档
│   └── sys.properties # 系统运行参数设定档
├── console.sh # 命令行工具
├── import.sh # 一键匯入指令碼或直譯式程式
├── logs
│   ├── import.log # 歸檔日誌
│   ├── job_stat.log # 任务状态记录
│   ├── ossimport2.log # 单机版作业记录
│   └── 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
│       └── localhost
│           ├── 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 # 成功启动并执行Task
│               └── A41506C07BF1DF2A3EDB4CE31756B93F_1499744514501@
├── localhost
│   ├── audit.log # Task作业记录，通过该日志可以查看错误原因
│   ├── DONE # Task运行成功标志
│   ├── error.list # Task错误清单，可以查看错误档案清单
│   ├── STATUS # 任务状态标志档案，内容为Failed或Completed，表示子任务失败或成功
│   └── TASK # Task描述资讯
├── worker # Worker正在启动并执行Task状态，运行完成后有Master管理
│   └── jobs
│       └── local_test
│           └── tasks

```

提示：

- Job运行资讯，可以查看log/ossimport2.log。
- Task的失败原因，可以查看master/jobs/\${JobName}/failed_tasks/\${TaskName}/audit.log。
- Task的失败档案，可以查看master/jobs/\${JobName}/failed_tasks/\${TaskName}/error.list。

常見錯誤及排除

請查看[常見錯誤及排除](#)。

3.3 分布式部署

下載

分布式部署目前只支援Linux，Windows暫不支援。

分布式版本下載地址ossimport-2.3.2.tar.gz，下載到本地後，使用命令 `tar -zxvf ossimport-2.3.2.tar.gz -C $HOME/ossimport` 解壓，解壓後的檔案結構如下：

```

ossimport
├── bin
│   ├── console.jar    # Console模組jar包
│   ├── master.jar    # Master模組jar包
│   ├── tracker.jar   # Tracker模組jar包
│   └── worker.jar     # Worker模組jar包
├── conf
│   ├── job.cfg        # Job設定檔範本
│   ├── sys.properties # 系統運行參數設定檔
│   └── workers        # Worker列表
├── console.sh         # 命令列工具，目前支援只Linux
├── logs               # 日誌目錄
└── README.md         # 說明文檔，使用前請仔細閱讀

```

注意：

- **OSS_IMPORT_HOME**：OssImport的根目錄，預設為解壓命令中的目錄`$HOME/ossimport`，也可以通過命令`export OSS_IMPORT_HOME=<dir>`或修改系統設定檔`$HOME/.bashrc` 設定，推薦使用預設目錄；
- **OSS_IMPORT_WORK_DIR**：OssImport的工作目錄，通過`conf/sys.properties`的配置項 `workingDir`指定，推薦為`$HOME/ossimport/workdir`；
- **OSS_IMPORT_HOME**或**OSS_IMPORT_WORK_DIR**請使用絕對路徑配置，如`/home/<user>/ossimport` 或 `/home/<user>/ossimport/workdir`。

配置

分布式部署有三個設定檔`conf/sys.properties`、`conf/job.cfg`、`conf/workers`。

- `conf/job.cfg`：分布式模式下任務的設定檔範本，資料移轉前請按照實際參數修改；
- `conf/sys.properties`：系統運行參數設定檔，如工作目錄、Worker運行參數等請在該檔案中配置；
- `conf/workers`：worker列表。

注意：

- 請在提交任務前確認`sys.properties`和`job.cfg`中的參數，任務提交後參數無法再修改；
- Worker 列表`workers`請啟動服務前確定，啟動後無法再增加或刪除。

運行

· 執行遷移任務

分布式部署時，執行任務的一般步驟是 修改任務設定檔、部署服務、清除同名任務、提交任務、啟動遷移服務、查看任務狀態、重試失敗子任務以及停止遷移任務。詳細說明如下：

- 部署服務。執行Linux終端執行`bash console.sh deploy`。



说明:

部署前請保證設定檔`conf/job.cfg`、`conf/workers` 已經修改完成。

- 清除同名任務。如果運行過同名任務，需要從新執行任務，請先清除同名任務。如果沒有運行過，或需要重試失敗任務，不需要執行清除命令。Linux終端中執行`bash console.sh clean job_name`。
- 提交資料移轉任務。OssImport不能重複提交同名任務，如果有請請使用`clean`命令清除。提交任務需要指定任務的設定檔，任務的設定檔範本在`conf/job.cfg`，建議在模板的基礎上修改。Linux終端執行`bash console.sh submit [job_cfg_file]`，提交設定檔為`job_cfg_file`的任務，`job_cfg_file`為選擇性參數，不指定是預設為`$OSS_IMPORT_HOME/conf/job.cfg`，`$OSS_IMPORT_HOME`預設為`console.sh`所在的目錄。
- 啟動服務。Linux終端執行`bash console.sh start`。
- 查看任務狀態。Linux終端執行`bash console.sh stat`。
- 失敗Task重試。由於網路或其它原因，Task可能運行失敗。失敗重試只重試失敗的Task，成功的Task不會重試。Linux下在終端執行`bash console.sh retry [job_name]`，`job_name`為選擇性參數，指定時重試任務`job_name`的失敗子任務，不指定`job_name`時重試所有任務的失敗子任務。
- 停止服務。Linux終端執行`bash console.sh stop`。

提示:

- `bash console.sh`在參數錯誤時，會自動提示命令格式；
- 設定檔和提交任務中的目錄，推薦使用絕對目錄；
- 任務的配置，即`job.cfg`中的配置項。



说明:

配置項提交後不能修改，請在提交前確定。

· 常見任務失敗原因

- 上傳過程中來源目錄的檔案發生了修改，log/audit.log裡會提示SIZE_NOT_MATCH相關字樣的錯誤，這種情況下老的檔案已經上傳成功，新的修改沒有上傳到OSS；
- 源檔案在上傳過程中被刪除，導致下載的時候失敗；
- 源檔案名不符合OSS命名規範（不能以/開頭，不能為空），導致上傳到OSS失敗；
- 下載資料來源檔案失敗；
- 程式異常退出，任務狀態為 Abort，這種情況請聯繫我們。

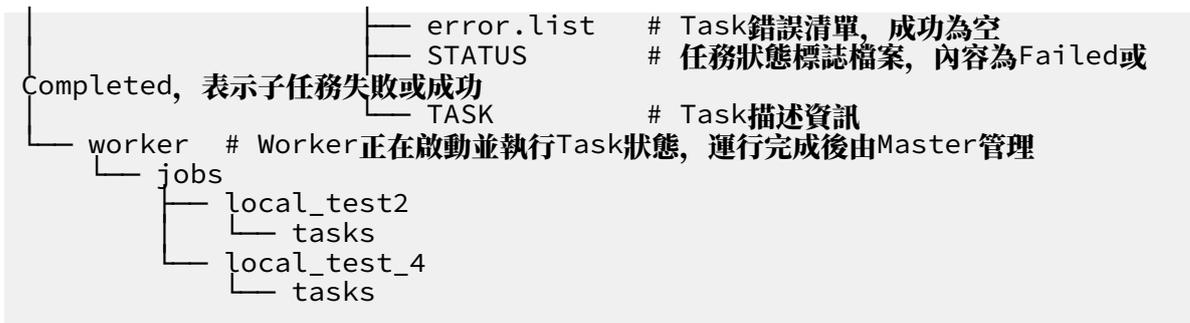
· 任務狀態及日誌

任務提交後，Master分解成Task，有Worker執行Task，Tracker收集Task狀態。任務運行完成後workdir目錄內容如下：

```

workdir
├── bin
│   ├── console.jar # Console模組jar包
│   ├── master.jar # Master模組jar包
│   ├── tracker.jar # Tracker模組jar包
│   └── worker.jar # Worker模組jar包
├── conf
│   ├── job.cfg # Job設定檔範本
│   ├── sys.properties # 系統運行參數設定檔
│   └── workers # Worker列表
├── logs
│   ├── import.log # 歸檔日誌
│   ├── master.log # Master日誌
│   ├── tracker.log # Tracker日誌
│   └── worker.log # Worker日誌
├── master
│   ├── jobqueue # 存放尚未分解完成的任務
│   ├── jobs # 存放任務運行狀態
│   └── xctooss # 任務名稱
│       ├── checkpoints # Master分解Job到Task的checkpoint點記
│       └── 錄
│           ├── 0
│           │   └── ED09636A6EA24A292460866AFDD7A89A.cpt
│           ├── dispatched # 已經分配給Worker尚未運行完成的Task
│           │   └── 192.168.1.6
│           └── failed_tasks # 運行失敗的Task
│               └── A41506C07BF1DF2A3EDB4CE31756B93F_1499348973217@
├── 192.168.1.6
│   └── 錯誤原因
│       ├── audit.log # Task作業記錄，通過該日誌可以查看
│       ├── DONE # Task運行成功標誌，失敗為空
│       └── error.list # Task錯誤清單，可以查看錯誤檔案清
├── 單
│   ├── STATUS # 任務狀態標誌檔案，內容為Failed或
│   │   └── Completed, 表示子任務失敗或成功
│   ├── TASK # Task描述資訊
│   ├── pending_tasks # 尚未分配的Task
│   └── succeed_tasks # 成功啟動並執行Task
│       └── A41506C07BF1DF2A3EDB4CE31756B93F_1499668462358@
├── 192.168.1.6
│   └── 誤原因
│       ├── audit.log # Task作業記錄，通過該日誌可以查看錯
│       └── DONE # Task運行成功標誌

```



提示:

- Job運行狀態查看logs/tracker.log, Worker的作業記錄查看logs/worker.log, Master的作業記錄查看 logs/master.log。
- Task的失敗原因, 可以查看master/jobs/\${JobName}/failed_tasks/\${TaskName}/audit.log。
- Task的失敗檔案, 可以查看master/jobs/\${JobName}/failed_tasks/\${TaskName}/error.list。

常見錯誤及排除

請查看常見錯誤及排除。

3.4 資料移轉

本文主要介紹OssImport在典型場景下資料移轉需求的實現。

OssImport介紹

部署方式

OssImport有單機模式和分布式模式兩種部署方式。

- 對於小於30TB的小規模資料移轉, 單機模式即可完成。
- 對於大規模的資料移轉, 請使用分布式模式。

分時限流

OssImport是基於master-worker的分布式架構。Worker有限流功能, 通過修改設定檔sys.properties的配置項workerMaxThroughput(KB/s)實現, 修改後需要重啟服務才能生效。

分布式部署情況下, 需要修改每個Worker的\$OSS_IMPORT_WORK_DIR/conf下的sys.properties, 然後重啟服務。

要實現分時限流, 可通過crontab定時修改sys.properties, 然後重啟服務生效。

添加Worker

Worker列表請在提交任務前確定，目前不支援動態添加。

只校驗不遷移數據

OssImport支援只校驗數據不遷移數據，設定任務設定檔`job.cfg`或`local_job.cfg`的配置項`jobType`為`audit`而不是`import`，其它配置與數據遷相同。

資料移轉增量模式

資料移轉增量模式，是指資料移轉任務啟動後，先進行一次全量遷移，每隔一段時間自動的進行增量資料移轉。第一次資料移轉任務為全量遷移，提交任務後立即啟動；後面的增量資料移轉每隔一個周期啟動一次。資料移轉增量模式適用於資料備份和資料同步。

增量模式有兩個配置項：

- `job.cfg` 中的`isIncremental`，表示是否開啟增量移轉模式，`true`表示開啟增量模式，`false`表示關閉增量模式，預設關閉。
- `job.cfg` 中的`incrementalModeInterval`，表示增量模式下的同步間隔，即增量資料移轉的間隔周期，單位秒。`isIncremental=true`時有效。可配置的最小值為900秒，不建議配置成小於3600秒的值，會浪費大量請求，造成額外的系統開銷。

指定遷移檔案的過濾條件

遷移檔案的過濾條件，即只遷移滿足特定條件的檔案。OssImport支援指定首碼和最後修改時間：

- `job.cfg` 中的`srcPrefix`，用來指定遷移檔案的首碼，預設為空。
 - 如果`srcType=local`，填寫本地目錄，需要完整路徑，以/進行分割並且以/結尾，如`c:/example/`或`data/example/`。
 - 如果`srcType`為`oss`、`qiniu`、`bos`、`ks3`、`youpai`、`s3`，則為待同步object的首碼，不包括bucket名稱，如`data/to/oss/`。遷移所有檔案時`srcPrefix`設定為空。
- `job.cfg` 中的`importSince`，用來指定遷移檔案的最後修改時間，整形，單位秒。`importSince`為Unix時間戳記，即自1970年1月1日UTC零點以來的秒數，通過命令`date +%s`獲取；預設值0，表示遷移全部數據。增量模式下只對第一次全量遷移有效，非增量模式對整個遷移任務有效。
 - 如果檔案的最後修改(`LastModified Time`)在`importSince`之前（包含）將被遷移。
 - 如果檔案的最後修改(`LastModified Time`)在`importSince`之後將不被遷移。

典型場景

場景1：從第三方儲存服務無縫切換到OSS

以下步驟可以完成從第三方儲存到OSS的無縫切換：

1. 全量遷移數據，此時業務仍在第三方儲存上，記下資料移轉的開始時間 T1，注意該時間為 Unix 時間戳記，即自 1970 年 1 月 1 日 UTC 零點以來的秒數，通過命令 `date +%s` 獲取。
2. 開啟 OSS 鏡像回源功能，資料移轉完成後，請在 OSS 控制台 設定服務 Bucket 的 **鏡像回源** 功能，回源地址為第三方儲存。
3. 讀寫切換到 OSS，此時 T1 前的數據從 OSS 讀取，T1 後的數據利用鏡像回源從第三方服務讀取，新數據完全寫入 OSS。
4. 增量資料移轉，增量資料移轉任務的設定檔 (`job.cfg` 或 `local_job.cfg`) 的配置項 `importSince=T1`，增量資料移轉完成的時間為 T2。



说明:

增量資料移轉，並非資料移轉的增量模式。

5. 刪除第三方儲存，T2 後，您業務的所有的讀寫都在 OSS 上，第三方儲存只是一份歷史數據，您可以根據需要決定保留或刪除。OssImport 負責數據的遷移和校驗，不會刪除任何數據。

場景2：本機資料遷移到 OSS

本機資料遷移到 OSS 的工具選擇：

- 對於小於 30TB 的數據從本地檔案或可以掛載到本地檔案系統的遷移情況，推薦使用 *OssUtil*，該工具簡單方便，OssUtil 支援檔案等級的增量上傳，通過 `-u/--update` 和 `--snapshot-path` 選項實現，詳細說明請使用 `ossutil help cp` 查看。
- 大規模資料移轉，請使用分布式版本的 *OssImport*。



说明:

本機資料的增量遷移時，檔案系統某些操作不會修改檔案的最後修改時間，比如 Windows 的 `cp`、`mv`，Linux 的 `mv`、`rsync` 帶 `-t` 或 `-a` 選項，這些操作的數據修改都不會被檢測到，也不會同步到 OSS。

場景3：OSS 之間的資料移轉

- 什麼時候使用 OssImport：
 - 不同區域間的 OSS 資料同步，推薦使用跨區域複製功能，該功能請在控制台上設定。
 - 由於安全原因，沒有開通跨區域複製的地域，可以使用 OssImport 遷移或備份數據。
 - 同一區域內，不同帳號、不同 Bucket 的資料移轉。
 - OSS 直接的資料移轉，推薦使用阿里雲內網，即使用 ECS、OSS 的網域名稱帶 `internal`。

- OSS直接資料移轉收費：
 - 如果使用了帶internal的網域名稱，不會產生流量費用，只有請求和儲存費用。
 - 如果沒有帶internal的網域名稱，會產生流量費用，具體請參看[計費](#)。
- 不推薦使用場景：
 - 開通了跨區域複製服務的地域之間的資料同步。
 - 利用增量模式在OSS之間同步檔案修改操作，OssImport只能同步檔案的修改操作(put/append/multipart)，不能同步讀取和刪除操作，資料同步的及時性沒有具體的SLA保證，請慎重選擇。推薦使用[上傳回調](#)。

遷移說明

ECS與流量

對於從雲端（非本地）遷移到OSS，且頻寬資源不是很充足的用戶，建議購買隨用隨付的ECS進行遷移，ECS配置如下：

- 付費方式選擇隨用隨付。
- 地域選擇OSS對應的地域。
- 頻寬峰值選100M。

在配置遷移服務時，將targetDomain設為帶internal的內網網域名稱；如果源端也是OSS，將srcDomain也設為帶internal的內網網域名稱，可以省掉從OSS源端下載的流量費，僅收取OSS訪問次數的費用。

HTTP資料移轉到OSS

HTTP資料移轉任務需要配置的參數：

- job.cfg 中的srcType配置為srcType=http，請注意字元大小寫。
- job.cfg 中的httpListFilePath，指定的HTTP地址清單檔案，請使用絕對路徑指定，如c:/example/http.list、/root/example/http.list。一個完整的HTTP連結是127.0.0.1/aa/bb.jpg，不同的切分方法最後會導致上傳到oss的路徑會不一樣：

```
http://127.0.0.1/aa/ bb.jpg # 第一行
http://127.0.0.1/ aa/bb.jpg # 第二行
```

第一行的檔案匯入到OSS後的的檔案名為destPrefix + bb.jpg，第二行的檔案名為destPrefix + aa/bb.jpg。httpPrefixColumn 指定網域名稱列，預設第一列，如上述的

127.0.0.1/aa/或127.0.0.1/。 `relativePathColumn` 指定在OSS中檔案名，如上述的 `bb.jpg`或 `aa/bb.jpg`。如果檔案中有多列，如下：

```
http://127.0.0.1/aa/ bb/cc dd/ee ff.jpg
```

配置應該如下：`httpPrefixColumn=1`，`relativePathColumn=4`。

- `job.cfg` 中的 `destAccessKey`、`destSecretKey`、`destDomain`、`destBucket` 等OSS 的配置。

HTTP資料移轉子任務切分參數：

- `taskObjectCountLimit`，每個 Task 最大的檔案數，預設10000；
- `taskObjectSizeLimit`，每個 Task 最大數據量，HTTP資料移轉時該參數無效，原因是 Master切分Task 時，如果每個HTTP檔案都是源上獲取檔案大小，每個檔案都有一次HTTP 請求開銷，會影響子任務分配的效率，進而影響子任務的並發執行，降低遷移的效率。
- 網域名稱，`httpListFilePath` 指定的檔案中第一列，連續相同的網域名稱任務按照 `taskObjectCountLimit`的限制切分，連續不同的網域名稱切分成不同的 Task，這種做法的目的是為了更好的複用串連。比如：

```
http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/ import/test1.txt
http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/ import/test2.txt
http://mingdi-bj.oss-cn-beijing.aliyuncs.com/ import/test3.txt
http://mingdi-bj.oss-cn-beijing.aliyuncs.com/ import/test4.txt
```

`taskObjectCountLimit`大於2的情況下，會切分成2個Task，而以下情況會切分成4個

Task：

```
http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/ import/test1.txt
http://mingdi-bj.oss-cn-beijing.aliyuncs.com/ import/test3.txt
http://mingdi-hz.oss-cn-hangzhou.aliyuncs.com/ import/test2.txt
http://mingdi-bj.oss-cn-beijing.aliyuncs.com/ import/test4.txt
```

所以 `httpListFilePath`指定的HTTP地址清單檔案，請先按照網域名稱排序。

網路流量與參數配置

以下參數的配置與網路流量有關：

- `sys.properties`中的`workerTaskThreadNum`，表示Worker並發執行的任務數量，如果網路較差、並發大，會出現大量逾時錯誤，此時應該降低並發量，修改該配置項，並重啟服務。
- `sys.properties`中的`workerMaxThroughput(KB/s)`，表示 Worker 流量的上限，如果業務需要限流，比如源端流控控制、網路限制等情況。該參數的值應該小於機器的最大網路流量，並根據業務需要評估。

- `job.cfg` 中的 `taskObjectCountLimit`，每個Task 最大的檔案數，預設10000。該參數會影響Task的數量，數量過小無法實現有效並發。
- `job.cfg` 中的 `taskObjectSizeLimit`，每個Task最大數據量，預設1GB。該參數會影響Task的數量，數量過小無法實現有效並發。



说明:

- 設定檔參數請盡量在啟動遷移前確定。
- `sys.properties`中的參數修改後，重啟遷移伺服器後才能生效。
- `job.cfg`任務提交後，任務的配置參數無法更改。

3.5 常見問題

- 1. `UnsupportedClassVersionError`異常

執行命令時異常:

```
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版本過低，更新到1.7或以上版本。

- 2. `InvocationTargetException`異常

使用submit命令提交任務報異常:

```
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：檢查是否將設定檔中原有的項刪除或者注釋掉了，不需要配置的項請在等號後面填空，不需要刪除。

• 3. too many open files

原因：`ulimit -n` 查看系統控制代碼。

- 如果值小於一萬以下的，可以通過 `ulimit -n 65536` 調大後，重啟進程即可；
- 如果本來就設定的比較大，那就不用 `sudo losf -n` 排查是哪些進程開啟了控制代碼。

• 4. windows 啟動後秒退

原因：大部分情況是java沒裝或者版本低於1.7導致的，或設定檔錯誤導致。

• 5. no jobs is running or finished

`submit` 命令提交完任務後，用 `stat` 查看任務狀態一直顯示：

```

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 剛提交，Master 需要先去掃描檔案清單，這時Task還沒有真正生成和分發，列印該日誌是正常的；
- 很長一段時間後依舊列印該錯誤，一般是沒有使用 `start` 命令啟動進程或者進程啟動後有異常退出。沒有啟動服務的情況，只需使用 `start` 就可以了；否則看下 `logs/ossimport.log`，找到異常原因並解決，然後再啟動服務進程。

• 6. stat 命令一直顯示 scanFinished: false

觀察Task的總數是否在增加：

- 如果Task總數增加，則表明Job的檔案清單還在List新的檔案。
- 如果Task總數沒變化，且Job 配置的是增量模式， `scanFinished` 不為true的情況下，會根據您配置的間隔時間定期掃描檔案清單，檢查有無新增或修改的檔案。
- 如果不是增量模式，Task數也不會增多，檢查日誌是否有異常。

- 7. 服務進程掛掉，日誌卻沒有輸出異常

原因：如果機器的可用記憶體是少於2GB 的，大機率是記憶體不足被殺掉了，檢查下dmsg日誌是否有記憶體不足被殺的記錄。

- 8. 進程掛掉或者殺掉進程後，重啟服務需要做什麼操作

直接調用start命令啟動服務即可，已經提交的Job不需要重新提交，只要不調用clean命令，所有提交過的Job都有斷點記錄，不會重做已經完成的工作。

- 9. 任務完成OSS控制台顯示的數據量比來源資料量小

Job全部成功上傳完後OSS控制台裡沒有看到bucket的大小有變化，或者和本地用du統計的大小相差很大。原因：OSS控制台的bucket數據量是延遲1小時更新的。du命令統計的是塊大小，會比實際檔案要大，可以參考如下命令統計本地目錄的真實大小：`ls -lR <目錄絕對路徑> | grep "\-rw" | awk '{sum+=$5}END{print sum}'`。

- 10. stat顯示的失敗任務如何處理

一般應該使用retry命令進行重試。

- 11. 有些失敗任務後反覆retry都無法成功

原因：查看檔案`$work_dir/master/jobs/$jobName/failed_tasks/$taskName/error.list`拿到失敗檔案的相對路徑，檢查該檔案是否有許可權訪問，是否被刪除，是否是軟串連，是否是亂碼的檔案名等。

- 12. 檔案名亂碼的檔案如何上傳到OSS

需要首先使用`export LANG="<your file name encode>"`，ls 查看該檔案名不亂碼之後，用clean命令清掉原來的Job，再用submit命令重新提交Job。

- 13. java.nio.file.AccessDeniedException

報異常：啟動服務時報java.nio.file.AccessDeniedException。原因：沒有許可權訪問設定檔目錄。

- 14. Task狀態顯示0個，但Job顯示完成

Task狀態顯示0個，但Job顯示完成，如下所示：

```
[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填寫錯誤，導致List不出來檔案；
 - srcPrefix下只有目錄，沒有檔案，因為目錄概念是OSS類比出來的，不會被真正上傳。
- **15. The bucket you are attempting to access must be addressed using the specified endpoint**

日誌報異常:

```
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填寫錯誤，請按照[網域名稱列表](#)填寫正確的網域名稱。

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

日誌報異常:

```
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: 檢查destAccessKey和destSecretKey是否填錯，詳細說明請參看[存取控制](#)。

- **17. InvocationTargetException**

submit命令提交任務時報異常:

```
submit job:/disk2/ossimport2/local_job.cfg
Exception in thread "main" java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
```

```

    at java.lang.reflect.Method.invoke(Method.java:606)
    at com.simontuffs.onejar.Boot.run(Boot.java:306)
    at com.simontuffs.onejar.Boot.main(Boot.java:159)
Caused by: java.lang.NullPointerException
    at com.aliyun.ossimport2.OSSImport2.doSubmitJob(OSSImport2.
java:289)
    at com.aliyun.ossimport2.OSSImport2.main(OSSImport2.java:120
)
    ... 6 more

```

原因：檢查conf/sys.properties中的配置項 workingDir 是否配置，配置是否正確；檢查設定檔路徑是否路徑正確。

- 18. 是否支援設定代理

不支援

- 19. OSS遷移到OSS為什麼還會產生費用

參考[訪問網域名稱](#)裡的網域名稱幫助，配置內網網域名稱後，將不收取流量費用，但訪問次數的費用是依舊計費的。

- 20. 同步的過程中顯示源端的檔案不存在

原因：Master是先 List 出檔案清單，之後按照檔案清單做資料移轉。當 List 完之後，源端某些檔案被刪除，就會出現源端檔案不存在的情況。這種檔案會被跳過，並把檔案輸出在錯誤清單裡。

- 21. 開啟增量模式，本地刪除後OSS上會不會被刪

開啟增量模式，本地刪除後OSS上會不會被刪，刪除操作不會被同步。

- 22. 開啟增量模式，有些新增的檔案沒有被同步

增量模式是採用對比檔案最後修改來判斷檔案是否為增量的，Linux的mv，Windows的cp，mv以及rsync帶-t或者-a參數等操作是不會修改檔案的最後修改時間的，這些操作修改的檔案是不會被掃描到的。

- 23. 又拍遷移的任務數一直顯示0

原因：又拍的比較複雜，主要分兩種情況：

```

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

```

，這條日誌如果recentLatency=30000這種一般就是在正常 List 的，又拍List 比較慢一般都是跑滿30秒逾時，30秒 List 出幾個檔案就返回幾個檔案過來，這種情況等任務慢慢List 出來就正常了；

- recentLatency很小，這種情況一般就是帳號密碼填錯之類的，因為又拍的sdk出錯只返回null，並不返回錯誤結果，所以只能通過抓包的方式獲得又拍返回的錯誤碼排查。

- 24. 又拍遷移時srcAccessKey和srcSecretKey填寫什麼
填寫又拍的操作員帳號和密碼。
- 25. 又拍遷移時一直顯示http 429錯誤
又拍對sdk訪問間隔做了限制，如果訪問稍微快一點，會進行限速，請聯繫又拍的客服放開限制。OssImport本身會對這種情況進行重試的。
- 26. 執行時unknown command “java”， unknown command “nohup” 之類的提示
原因：使用的命令沒有安裝，請用yum或apt-get或zypper 等命令安裝相應的命令。

- 27. 任務與設定檔不符

Job設定檔看上去是對的，但跑的時候看上去明顯和Job設定檔配置的不一樣。只有sys.properties是改完之後重啟就生效的，Job的設定檔一旦提交後，修改是不會生效的，需要clean 掉原有的Job，然後重新submit新的設定檔。

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

日誌報異常：

```
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：檢查配置項destBucket是否填寫正確，bucket是不帶 / 以及其它路徑的。

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

日誌報異常：

```
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，一般都是網路打滿的時出現的，OssImport內部會重試，如果重試完後依舊失敗，可以在任務完成後調用retry命令再次重試。

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

日誌報異常：

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

原因：非ieECS的機器不能使用帶internal的網域名稱。

- 31. The specified bucket is not valid

日誌報異常：

```
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：設定檔裡的destDomian配置的網域名稱不能帶bucket名稱。

- 32. 設定檔裡的srcPrefix能單獨指定檔案嗎

不可以，srcPrefix只支援目錄或者首碼等級，單個檔案上傳可以用別的更簡單的工具。

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

日誌報異常：

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

原因：

- 本地機器時間不對，與伺服器時間相差15分鐘以上，該情況居多。
- 可能是並發太高，尤其是CPU佔用率很高，導致在並發上傳慢。

- 34. No route to host

日誌裡顯示錯誤No route to host，一般是本地防火牆或者 iptables 等原因導致網路不通。

- 35. Unknown http list file format

使用 http 模式日誌顯示該錯誤，是因為指定的http列表檔案格式不對：

- 從其它系統上拷過來的檔案，可以用mac2unix, dos2unix等相關命令轉化檔案格式；
- 檔案裡有某些行不符規則，比如某行少於兩列。

- 36. The boject key "/xxxxx.jpg" is invalid

日誌報異常：

```
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：

- 檢查srcPrefix是否是作為目錄的但沒有以/ 結尾；
- 檢查destPrefix是否以/ 或者\ 開頭了。

4 RAM策略编辑器

地址

[RAM Policy Editor](#)

使用

RAM授權策略由若干條規則群組成，使用RAM策略编辑器，可以在介面上逐條添加/刪除規則，並自動建置原則的JSON文本。用戶添加完所有規則後，只需要將JSON文本拷貝，然後粘貼到存取控制（RAM）控制台的建立授權策略內容框內。

具體操作請參見[建立自訂授權策略](#)。

RAM策略编辑器中，每條規則需要設定其Effect、Actions、Resources和Conditions：

- Effect

指定這條規則是允許訪問（Allow）還是禁止訪問（Deny）。

- Actions

指定訪問資源的動作，可以選擇多項。一般來說用戶使用提供的通配動作就足夠了：

- `oss:*`表示允許所有動作。
- `oss:Get*`表示允許所有的讀動作。
- `oss:Put*`表示允許所有的寫動作。

更多資訊請參見[RAM Policy Editor README](#)。

• Resources

指定授權訪問的OSS的資源，可以指定多個，每個是以下形式：

- 表示某個bucket：`my-bucket`（此時對bucket下的檔案沒有許可權）
- 表示某個bucket下面所有檔案：`my-bucket/*`（此時對bucket本身沒有許可權，例如ListObjects）
- 表示某個bucket下某個目錄：`my-bucket/dir`（此時對dir/下面的檔案沒有許可權）
- 表示某個bucket下某個目錄下面所有檔案：`my-bucket/dir/*`（此時對dir沒有許可權，例如ListObjects）
- 填寫完整的資源路徑：`acs:oss:*:1234:my-bucket/dir`，其中1234為用戶的User ID（在控制台查看）

EnablePath

當用戶需要對某個目錄授權時，往往還需要保證對上一層目錄也有List許可權，例如用戶對`my-bucket/users/dir/*`賦予讀寫權限，為了在控制台（或其他工具）能夠查看這個目錄，用戶還需要以下許可權：

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

勾選EnablePath選項時，上面這些許可權會自動添加。

• Conditions

指定授權訪問時應該滿足的條件，可以指定多個。

例子

授權對`my-bucket`及其檔案全部的許可權：

更多例子請參見[RAM Policy Editor](#)。

5 ossftp

5.1 如何快速安裝OSS FTP

簡介

OSS FTP工具是一個特殊FTP server, 它接收普通FTP請求後, 將對檔案、檔案夾的操作映射為對OSS的操作, 從而使得您可以基於FTP協議來管理儲存在OSS上的檔案。



说明:

生產環境請使用oss sdk, OSS FTP工具主要面向個人用戶使用。

· 主要特性

- 跨平台: 無論是Windows、Linux還是Mac, 無論是32位還是64位作業系統, 無論是圖形介面還是命令列都可以運行。
- 免安裝: 解壓後可直接運行。
- 免設定: 無需設定即可運行。
- 透明化: FTP工具是python寫的, 您可以看到完整的源碼, 我們稍後也會開源到Github。

· 主要功能

- 支援檔案和檔案夾的上傳、下載、刪除等操作。
- 通過Multipart方式, 分區上傳大檔案。
- 支援大部分FTP指令, 可以滿足日常FTP的使用需求。



说明:

- 1. 目前在1.0版本中, 考慮到安裝部署的簡便, OSS FTP工具沒有支援TLS加密。由於FTP協議是明文傳輸的, 為了防止您的密碼洩漏, 建議將FTP server和client運行在同一台機器上, 通過127.0.0.1:port的方式來訪問。
- 2. 不支援rename和move操作。
- 3. 安裝包解壓後的路徑不要含有中文。
- 4. FTP server的管理控制頁面在低版本的IE中可能打不開。
- 5. FTP server支援的Python版本: Python2.6和Python2.7。

下載

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

由於Windows不會預設安裝Python2.7，所以安裝包中包含了Python2.7，免去您python安裝配置的麻煩，解壓即可使用。

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

由於Linux/Mac系統預設會安裝Python2.7或Python2.6，所以安裝包中不再包含可執行檔python，只包含了相關依賴庫。

運行

首先解壓之前下載的檔案，然後根據環境情況選擇不同的運行方式。

- Windows: 雙擊運行start.vbs即可
- Linux: 開啟終端，運行

```
$ bash start.sh
```

- Mac: 雙擊start.command，或者在終端運行

```
$ bash start.command
```

上述步驟會啟動一個FTP server，預設監聽在127.0.0.1的2048通信埠。同時，為了方便您對FTP server的狀態進行管控，還會啟動一個web伺服器，監聽在127.0.0.1的8192通信埠。如果您的系統有圖形介面，還會自動開啟控制頁面，如下所示：



说明:

大部分情況不要任何配置，就可以運行一個FTP server了，如果想對FTP server進行配置，請注意需要重啟才能生效。

串連到FTP server

推薦使用[FileZilla](#)客戶端去串連FTP server。下載安裝後，按如下方式串連即可：

- 主機: 127.0.0.1
- 登入類型: 正常
- 用戶: access_key_id/bucket_name
- 密碼: access_key_secret



说明:

- 用戶中，/是必須的，如用戶tSxyixxxxxxwPMEp/test-hz-jh-002。

- `access_key_id`和`access_key_secret`的獲取，請參見[OSS存取控制](#)。

進階使用

- 通過控制頁面管理FTP server

- 修改監聽地址

如果需要通過網路來訪問FTP server，那麼需要修改監聽地址，因為預設的監聽地址127.0.0.1隻允許來自本地的訪問。可以修改成內網ip或公網ip。

- 修改監聽通信埠

修改FTP server監聽的通信埠，建議通信埠大於1024，因為監聽1024以下的通信埠時需要管理員權限。

- 修改日誌等級

設定FTP server的記錄層級。FTP server的日誌會輸出到`data/ossftp/`目錄下，可以通過控制頁面的日誌按鈕線上查看。預設的日誌界別為INFO，列印的日誌資訊較少，如果需要更詳細的日誌資訊，可以修改為DEBUG模式。如果希望減少日誌的輸出，可以順位為WARNING或ERROR等。

- 設定Bucket endpoints

FTP server預設會探索bucket的所屬location資訊，隨後將請求發到對應的region（如`oss-cn-hangzhou.aliyuncs.com`或`oss-cn-beijing.aliyuncs.com`），FTP server會優先嘗試內網訪問oss。如果您設定了bucket endpoints，如設定為`test-bucket-a.oss-cn-hangzhou.aliyuncs.com`，那麼當訪問`test-bucket-a`時，就會使用`oss-cn-hangzhou.aliyuncs.com`網域名稱。

- 設定顯示語言

通過設定`cn/en`，可修改FTP控制頁面的顯示語言為中文/英文。



说明:

- 所有修改都需要重啟才能生效。
- 上述的所有修改其實都是修改的ftp根目錄下的`config.json`，所以您可以直接修改該檔案。

- 直接啟動FTP server(Linux/Mac)

可以直接啟動`ossftp`目錄下的`ftpserver.py`，免去`web_server`的開銷。

```
$ python ossftp/ftpserver.py &
```

配置修改方式同上。

可能遇到的問題

- 如果串連FTP server時，遇到以下錯誤：

有兩種可能：

- 輸入的 `access_key_id` 和 `access_key_secret` 有誤。

解決：請輸入正確的資訊後再重試。

- 所用的 `access_key` 資訊為 ram 子賬戶的 `access_key`，而子賬戶不具有 List buckets 許可權。

解決：當使用子賬戶訪問時，請在控制頁面中指定 bucket endpoints，即告訴 FTP server 某個 bucket 應該用什麼 endpoint 來訪問。同時，子賬戶也需要一些必須的許可權，關於使用 ram 訪問 oss 時的存取控制，請參考文檔 [存取控制](#)。具體如下。

■ 只讀訪問

OSS FTP 工具需要的許可權列表為 ListObjects、GetObject、HeadObject。關於如何建立一個具有隻讀訪問的 ram 子賬戶，請參考圖文教程 [如何結合 ram 實現檔案分享權限設定](#)。

■ 上傳檔案

如果允許 ram 子賬戶上傳檔案，還需要 PutObject。

■ 刪除檔案

如果允許 ram 子賬戶刪除檔案，還需要 DeleteObject。

- 如果您在 Linux 下運行 FTP server，然後用 FileZilla 串連時遇到如下錯誤：

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

一般是因為本地的中文編碼有問題。在將要運行 start.sh 的終端中輸入下面的命令，然後再重新啟動即可。

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

5.2 Discuz 如何儲存遠程附件到 OSS

前言

網站遠程附件功能是指將用戶上傳的附件直接儲存到遠端的儲存伺服器，一般是通過 FTP 的方式儲存到遠端 FTP 伺服器。

目前 Discuz 論壇、phpwind 論壇、Wordpress 個人網站等都支援遠程附件功能。

本文介紹如何基於Discuz論壇儲存遠程附件。

準備工作

申請OSS帳號，並且建立一個public-read的bucket。這裡需要許可權為public-read是因為後面需要匿名訪問。

詳細步驟

測試所用Discuz版本為Discuz! X3.1，以下是詳細的設定流程：

1. 登入Discuz網站，進入管理介面後，先點擊全域，再點擊上傳設定，如下圖所示。
2. 選擇遠程附件，然後開始設定。



说明:

- 啟用遠程附件，請選擇是。
- 啟用SSL串連，請選擇否。
- FTP伺服器位址，即運行ossftp工具的地址，通常填寫 127.0.0.1 即可。
- FTP服務的連接埠號碼，預設為 2048。
- FTP登入用戶名，格式為 AccessKeyID/BucketName。注意這裡的 / 不是或的意思。
- FTP的登入密碼，為 AccessKeySecret。
- 被動模式串連，選擇預設的是即可。



说明:

- 遠程附件目錄，填 . 表示在Bucket的根目錄下建立上傳目錄。
- 遠端存取URL，填 `http://BucketName.Endpoint` 即可。



说明:

這裡測試所用bucket為test-hz-jh-002，屬於杭州區域的，所以這裡填寫的是`http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com`。注意BucketName要和Endpoint匹配。

- 逾時時間，設定為0即可，表示服務預設。
- 完成設定後，可以點擊測試遠程附件，如果成功則會出現如下畫面。

3. 發帖驗證

發貼時上傳圖片附件如下所示。

在圖片上右鍵點擊，選擇在“新建標籤頁中開啟圖片”，如下所示。

這就表示圖片已經上傳到了OSS的test-hz-jh-002中。

5.3 Phpwind如何儲存遠程附件到oss

前言

網站遠程附件功能是指將用戶上傳的附件直接儲存到遠端的儲存伺服器，一般是通過FTP的方式儲存到遠端FTP伺服器。

目前Discuz論壇、phpwind論壇、Wordpress個人網站等都支援遠程附件功能。

本文介紹如何基於Phpwind論壇儲存遠程附件。

準備工作

申請OSS帳號，並且建立一個public-read的bucket。這裡需要許可權為public-read是因為後面需要匿名訪問。

詳細步驟

測試所用版本為phpwind8.7，以下為詳細設定流程。

1. 登入網站

進入管理介面，依次選擇全域 > 上傳設定 > 遠程附件。

2. 開始設定



说明:

- 使用FTP上傳，請選擇開啟。
- 網站附件地址，填寫<http://bucket-name.endpoint>這裡測試所用bucket為test-hz-jh-002，屬於杭州區域的，所以這裡填寫的是<http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com>，注意BucketName要和Endpoint匹配。
- FTP伺服器位址，即運行ossftp工具的地址，一般填127.0.0.1即可。
- FTP服務的連接埠號碼，預設為2048。
- FTP上傳目錄，預設填.即可，表示在bucket的根目錄開始建立附件目錄。
- FTP登入用戶名，格式為AccessKeyID/BucketName。注意這裡的/不是或的意思。

- FTP的登入密碼，為AccessKeySecrete關於AccessKeyID和AccessKeySecrete的獲取，可以登入阿里雲控制台的Access Key管理進行查看。
- FTP逾時時間，設定為10就表示如果10秒內請求沒有返回結果就會逾時返回。

3. 發帖驗證

phpwind不能在設定完成後直接點擊測試，我們這裡髮帶圖片的文章來驗證下。

在圖片點擊右鍵，在新建標籤頁中開啟圖片，可以看到下圖。

通過圖中的URL，我們可以判斷圖片已經上傳到了OSS的test-hz-jh-002 Bucket中。

5.4 Wordpress如何儲存遠程附件到oss

前言

網站遠程附件功能是指將用戶上傳的附件直接儲存到遠端的儲存伺服器，一般是通過FTP的方式儲存到遠端FTP伺服器。

目前Discuz論壇、phpwind論壇、Wordpress個人網站等都支援遠程附件功能。

本文介紹如何基於Wordpress論壇儲存遠程附件。

準備工作

申請OSS帳號，並且建立一個public-read的bucket。這裡需要許可權為public-read是因為後面需要匿名訪問。

詳細步驟

wordpress本身是不支援遠程附件功能的，但是可以通過第三方的外掛程式來做遠程附件。作者所用wordpress版本為4.3.1，所用外掛程式為Hacklog Remote Attachment，以下為具體設定步驟。

1. 登入wordpress網站，選擇安裝外掛程式，搜關鍵詞FTP，選擇Hacklog Remote Attachment安裝。

2. 設定。

- FTP伺服器位址，即運行ossftp工具的地址，一般填127.0.0.1即可。
- FTP服務的連接埠號碼，預設為2048。
- FTP登入用戶名，格式為AccessKeyID/BucketName，注意這裡的/不是或的意思。
- FTP的登入密碼為AccessKeySecrete。



说明:

關於AccessKeyID和AccessKeySecrete的獲取，可以登入阿里雲控制台的Access Key管理進行查看。

- FTP逾時時間，預設設定為30秒即可。
- 遠程基本URL填 `http://BucketName.Endpoint/wp`。這裡測試所用bucket為test-hz-jh-002，屬於杭州區域的，所以這裡填寫的是 `http://test-hz-jh-002.oss-cn-hangzhou.aliyuncs.com/wp`。
- FTP遠程路徑，填wp表示所有附件都會儲存在bucket的wp目錄下。注意6和7要對應起來。
- HTTP遠程路徑，填.即可。

具體資訊見下圖。

3. 驗證。

設定好之後，點擊保存的同時，會做測試，測試結果會在頁面上方顯示，如下圖所示表示測試成功。

4. 發布新文章，並插入圖片。

現在開始寫一篇新文章，並測試遠程附件。建立好文章後，點擊添加媒體來上傳附件。

上傳附件如下圖所示。

5. 上傳完附件，點擊發布，即可看到文章了。

仍然通過右鍵點擊圖片，通過新建連結來開啟圖片即可看到圖片的URL如下圖所示。

通過圖片的URL，我們可以判定圖片已經成功上傳到了OSS。

5.5 如何結合RAM實現檔案分享權限設定

簡介

本文主要介紹如何結合RAM服務，共用用戶bucket中的檔案/檔案夾，讓其他用戶只讀，而bucket的owner可以修改。

思路就是：開通RAM -> 新建只讀授權策略 -> 建立子用戶 -> 向子用戶授權 -> FTP登入驗證

獲取賬戶ID

獲取你的account ID。具體參考下圖步驟。


```
}
```

將上面的*****替換為你自己的賬戶ID，test-hz-john-001替換為你自己的bucket名，然後整體拷貝到策略內容裡面，最後點擊新建授權策略即可。

建立用戶

上面的授權策略生命了一種只讀策略，下面新建一個用戶並給予他這種策略。先是建立用戶，步驟如下：



说明:

注意保存新用戶的access_key。

給用戶授權

下面將之前建立的策略授權給該用戶。

用子賬戶登入

用子賬戶的access_key和之前授權策略中的bucket登入，即可下載檔案夾或檔案，但上傳會失敗。

6 ossfs

6.1 快速安裝

ossfs 能讓您在Linux系統中把OSS bucket 掛載到本地檔案系統中，您能夠便捷地通過本地檔案系統操作OSS 上的對象，實現數據的共用。

主要功能

ossfs 基於s3fs 構建，具有s3fs 的全部功能。主要功能包括：

- 支援POSIX 檔案系統的大部分功能，包括檔案讀寫，目錄，連結操作，許可權，uid/gid，以及擴充屬性（extended attributes）
- 通過OSS 的multipart 功能上傳大檔案。
- MD5 校驗保證資料完整性。

局限性

ossfs提供的功能和性能和本地檔案系統相比，具有一些局限性。具體包括：

- 隨機或者追加寫檔案會導致整個檔案的重寫。
- 元數據操作，例如list directory，性能較差，因為需要遠端存取OSS伺服器。
- 檔案/檔案夾的rename操作不是原子的。
- 多個客戶端掛載同一個OSS bucket時，依賴用戶自行協調各個客戶端的行為。例如避免多個客戶端寫同一個檔案等等。
- 不支援hard link。
- 不適合用在高並發讀/寫的場景，這樣會讓系統的load升高。

安裝及使用

- 安裝包下載

Linux發行版	下載
Ubuntu 16.04 (x64)	ossfs_1.80.5_ubuntu16.04_amd64.deb
Ubuntu 14.04 (x64)	ossfs_1.80.5_ubuntu14.04_amd64.deb
CentOS 7.0 (x64)	ossfs_1.80.5_centos7.0_x86_64.rpm

Linux發行版	下載
CentOS 6.5 (x64)	ossfs_1.80.5_centos6.5_x86_64.rpm

由於低版本的Linux發行版本核心版本比較低，ossfs進程在運行過程中容易出現掉線或者其他問題，因此建議用戶將作業系統升級到CentOS 7.0或者Ubuntu 14.04以及以上版本。

- 安裝方法

- 對於Ubuntu，安裝命令為：

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

- 對於CentOS6.5及以上，安裝命令為：

```
sudo yum localinstall your_ossfs_package
```

- 對於CentOS5，安裝命令為：

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

- 使用方法

設定bucket name 和 AccessKeyId/Secret資訊，將其存放在/etc/passwd-ossfs 檔案中，注意這個檔案的許可權必須正確設定，建議設為640。

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

將OSS bucket mount到指定目錄。

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

樣本

將my-bucket這個bucket掛載到/tmp/ossfs目錄下，AccessKeyId是faint，AccessKeySecret是123，oss endpoint是http://oss-cn-hangzhou.aliyuncs.com

```
echo my-bucket:faint:123 > /etc/passwd-ossfs
chmod 640 /etc/passwd-ossfs
mkdir /tmp/ossfs
ossfs my-bucket /tmp/ossfs -ourl=http://oss-cn-hangzhou.aliyuncs.com
```



注意：

如果您使用在阿里雲購買的雲虛擬機器主機（ECS）來提供ossfs服務，您可以使用內網網域名稱，比如在這個例子您可以將oss endpoint 改成oss-cn-hangzhou-internal.aliyuncs.com，這樣可以節省頻寬方面的費用。OSS的內網網域名稱請參考[訪問網域名稱和資料中心](#)。

卸載bucket

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

更多詳細內容請參考：[GitHub ossfs](#)

版本日誌

請參考：[GitHub ChangeLog](#)

6.2 FAQ

- Q: ossfs適合什麼樣的程式?
 - ossfs能把oss bucket掛載到本地，如果您使用的軟體沒有支援OSS，但您又想讓數據能自動同步到OSS，那麼ossfs是很好的選擇。
- Q: ossfs有什麼局限性?
 - 由於數據需要經過網路同步到雲端，ossfs在性能和功能上可能與本地檔案系統有差距。如果您想讓資料庫等對io要求很高的應用跑在ossfs掛載的盤上，請慎重考慮。和本地檔案系統具體差異：
 - 隨機或者追加寫檔案會導致整個檔案的重寫。
 - 元數據操作，例如list directory，性能較差，因為需要遠端存取OSS伺服器。
 - 檔案/檔案夾的rename操作不是原子的。
 - 多個客戶端掛載同一個oss bucket時，依賴用戶自行協調各個客戶端的行為。例如避免多個客戶端寫同一個檔案等等。
 - 不支援hard link。
- Q: ossfs一定要阿里雲的機器才能用麼?
 - ossfs不限制一定要阿里雲的內網才可以使用，外網機器依然可以使用。
- Q: ossfs能不能同時掛載多個OSS bucket?
 - 可以的，在passwd-ossfs檔案中寫入多個OSS配置資訊即可。支援不同帳號的OSS。
- Q: 我在yum/apt-get安裝ossfs，遇到conflicts with file from package fuse-devel的錯誤，請問是怎麼回事?
 - 您的系統中存在老版本的fuse，請先使用相關的包管理器卸載，再重新安裝ossfs。

- Q: ossfs工作不正常，如何debug?
 - 您可以使用在掛載時，加上-d -o f2參數，ossfs會把日誌寫入到系統日誌中。在centos系統中，在/var/log/messages中。
 - 您也可以掛載時使用-f -d -o f2參數，ossfs會把日誌輸出到螢幕上。
- Q: 為什麼我在mount時遇到 ossfs: unable to access MOUNTPOINT /tmp/ossfs: Transport endpoint is not connected這樣的錯誤?
 - 請先umount對應的目錄。
 - 請檢查您在使用ossfs掛載時，填入的url參數是否正確，是否和bucket/access key id/access key secret匹配。
 - 特別注意：url中不包含bucket的名字。例如：您在oss控制台中看到bucket的網域名稱是這樣的：`ossfs-test-1.oss-cn-hangzhou.aliyuncs.com`。那麼填入的url則是：`http://oss-cn-hangzhou.aliyuncs.com`。
- Q: ossfs提示ossfs: unable to access MOUNTPOINT /tmp/odat: No such file or directory
 - 這是您未建立該目錄導致的，在掛載前需要建立對應目錄。
- Q: 我把bucket掛載到本地後，ls目錄，卻收到operation not permitted錯誤，這是為什麼?
 - 請檢查您的bucket中，是否包含目錄名含有不可見字元的OSS object。檔案系統對檔案/目錄名有更嚴格的限制，因此會收到上述錯誤。使用其他工具對這些object重新命名後，ls就能正確顯示目錄內容了。
- Q: 我的一個目錄下有非常多的檔案，為什麼ls該目錄很慢?
 - 假設一個目錄下有n個檔案，那麼ls該目錄至少需要n次OSS http requests。在檔案非常多的時候，這可能造成嚴重的性能問題。
 - 您可以採用下面兩個辦法優化：
 - 通過-omax_stat_cache_size=xxx參數增大stat cache的size，這樣第一次ls會較慢，但是後續的ls就快了，因為檔案的元數據都在本地cache中。預設這個值是1000，大約消耗4MB記憶體，請根據您機器記憶體大小調整為合適的值。
 - 使用ls -f命令，這個命令會消除與OSS的n次http請求。
 - 具體參見[issue 13](#)
- Q: ossfs掛載時如何設定許可權?
 - 如果要允許其他用戶訪問掛載檔案夾，可以在運行ossfs的時候指定allow_other參數：
 - `ossfs your_bucket your_mount_point -ourl=your_endpoint -o`

```
allow_other
```

- 為什麼使用allow_other參數，仍然不能訪問檔案？
 - 注意：allow_other是賦予掛載目錄其他用戶訪問的許可權，不是裡面的檔案！如果您要變更檔案夾中的檔案，請用chmod命令。
- allow_other預設賦予掛載目錄777許可權，我想讓掛載目錄的許可權為770，該怎麼辦？
 - 可以通過umask來設定，參見[這裡](#)。
- Q: 如果要使掛載的檔案夾(/tmp/ossfs)屬於某個user:
 - 方法一：如果要使掛載的檔案夾(/tmp/ossfs)屬於某個user，則需要以user的身份建立掛載檔案夾和使用ossfs：
 - `sudo -u user mkdir /tmp/ossfs`
 - `sudo -u user ossfs bucket-name /tmp/ossfs`
 - 方法二：首先通過id命令獲得指定用戶的uid/gid資訊。例如獲取www用戶的uid/gid資訊：`id www`；然後掛載時指定uid/gid參數：
 - ```
ossfs your_bucket your_mountpoint -ourl=your_url -oid=your_uid
-ogid=your_gid
```
- 注意：uid/gid都是數字。
- Q: 我不是root用戶，如何umount ossfs掛載的目錄
  - `fusermount -u your_mountpoint`
- Q: 如何開機自動掛載ossfs？
  - Step 1 首先請參考使用說明，把bucket name, access key id/secret等資訊寫入/etc/passwd-ossfs，並將該檔案許可權修改為640。
    - ```
echo your_bucket_name:your_access_key_id:your_access_key_secret  
>  
/etc/passwd-ossfs
```
 - `chmod 640 /etc/passwd-ossfs`
 - Step 2 接下來針對不同的系統版本，設定方式有所不同
 - Step 2A 通過fstab的方式自動mount（適用於ubuntu14.04, centos6.5）
 - 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

- 其中上述命令中的your\_XXX資訊需要根據您的bucket name等資訊填入。
- 保存/etc/fstab檔案。執行mount -a命令，如果沒有報錯，則說明設定正常。
- 到這一步，ubuntu14.04就能自動掛載了。centos6.5還需要執行下面的命令：
  - `chkconfig netfs on`
- Step 2B 通過開機自啟動指令碼或直譯式程式mount（適用於centos7.0及以上的系統）
  - 在/etc/init.d/目錄下建立檔案ossfs，把模板檔案中的內容拷貝到這個新檔案中。並將其中的your\_XXX內容改成您自己的資訊。
  - 執行命令：`chmod a+x /etc/init.d/ossfs`
  - 上述命令是把新建立的ossfs指令碼或直譯式程式賦予可執行許可權。您可以執行該指令碼或直譯式程式，如果指令檔內容無誤，那麼此時oss中的bucket已經掛載到您指定的目錄下了。
  - 執行命令：`chkconfig ossfs on`
  - 上述命令是把ossfs啟動指令碼或直譯式程式作為其他服務，開機自動啟動。
  - 好了，現在ossfs就可以開機自動掛載了。總結起來，如果您是ubuntu14.04和centos6.5，您需要執行Step 1 + Step 2A；如果您是centos7.0系統，您需要執行Step 1 + Step 2B。
- Q: 遇到fusermount: failed to open current directory: Permission denied錯誤如何解決？
  - 這是fuse的一個bug，它要求目前使用者對目前的目錄（非掛載目錄）有讀許可權。解決的辦法就是cd到一個有讀許可權的目錄再運行ossfs命令
- Q: 我需要以www用戶掛載ossfs，此時如何設定開機自動掛載？
  - 參照上面的問題的解答，Step 1照做，對Step 2B稍加修改，修改/etc/init.d/ossfs中的命令為：

```
sudo -u www ossfs your_bucket your_mountpoint -ourl=your_url
```
  - 設定自啟動指令碼或直譯式程式中允許使用sudo，編輯/etc/sudoers，將其中的Defaults requiretty這行改為#Defaults requiretty（注釋掉）
- Q: 遇到fusermount: failed to open current directory: Permission denied錯誤如何解決？
  - 這是fuse的一個bug，它要求目前使用者對目前的目錄（非掛載目錄）有讀許可權。解決的辦法就是cd到一個有讀許可權的目錄再運行ossfs命令。

- Q: 使用ECS掛載ossfs, 如何避免因背景程式掃描檔案而產生費用?
  - 程式掃描ossfs掛載的目錄, 會轉換成向OSS的請求, 如果請求次數很多, 會產生費用 (1分錢/1萬次)。如果是`updatedb`, 可以通過修改`/etc/updatedb.conf`讓它跳過。具體做法是:
    1. 在`PRUNEFST` =後面加上`fuse.ossfs`
    2. 在`PRUNEPATHS` =後面加上掛載的目錄
  - 如何確定是哪個進程掃了我的目錄?
    1. 首先安裝auditd: `sudo apt-get install auditd`
    2. 啟動auditd: `sudo service auditd start`
    3. 設定監視掛載目錄: `auditctl -w /mnt/ossfs`
    4. 在auditlog中可以查看是哪些進程訪問了這個目錄: `ausearch -i | grep /mnt/ossfs`
- Q: 使用ossfs上傳到OSS的檔案Content-Type全是” application/octet-stream” 是怎麼回事?
  - ossfs通過查詢`/etc/mime.types`中的內容來確定檔案的Content-Type, 請檢查這個檔案是否存在, 如果不存在, 則需要添加:
    1. 對於ubuntu可以通過`sudo apt-get install mime-support`來添加
    2. 對於centos可以通過`sudo yum install mailcap`來添加
    3. 也可以手動添加, 每種格式一行, 每行格式為: `application/javascript js`
- Q: 如何使用supervisor啟動ossfs?
  1. 安裝supervisor, 在ubuntu中執行`sudo apt-get install supervisor`
  2. 建立一個目錄, 編輯ossfs的啟動指令碼或直譯式程式:

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

寫入如下數據:

```
卸載
fusermount -u /mnt/ossfs
重新掛載, 必須要增加-f參數運行ossfs, 讓ossfs在前台運行
exec ossfs my-bucket my-mount-point -ourl=my-oss-endpoint -f
```

3. 編輯`/etc/supervisor/supervisord.conf`, 在最後加入下面一段:

```
[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. 運行supervisor:

```
supervisord
```

#### supervisord

#### 5. 確認一切正常:

```
ps aux | grep supervisor # 應該能看到supervisor進程
ps aux | grep ossfs # 應該能看到ossfs進程
kill -9 ossfs # 殺掉ossfs進程，supervisor應該會重啟它，不要使用killall，因為killall發送SIGTERM，進程正常退出，supervisor不再去重新運行ossfs
ps aux | grep ossfs # 應該能看到ossfs進程
```

如果出錯，請檢查/var/log/supervisor/supervisord.log和/var/log/ossfs.log。

- Q: 遇到” fuse: warning: library too old, some operations may not work” 怎麼辦?

出現的原因是：ossfs編譯時間所使用的libfuse版本 比運行時連結到的libfuse版本高。這往往是用戶自行安裝了libfuse導致的。使用我們提供的rpm包安裝ossfs，無需再安裝libfuse。

在CentOS-5.x和CentOS-6.x上我們提供的rpm包裡包含了libfuse-2.8.4，如果在啟動並執行時候環境中有libfuse-2.8.3，並且ossfs被連結到了舊版本的fuse上，就會出現上述warning。

#### 1. 如何確認ossfs運行時連結的fuse版本?

- 運行`ldd $(which ossfs) | grep fuse`
- 例如結果是” /lib64/libfuse.so.2” ，那麼通過`ls -l /lib64/libfuse*`可以看到fuse的版本

#### 2. 如何讓ossfs連結到正確的版本?

- 首先通過`rpm -ql ossfs | grep fuse`找到libfuse的目錄
- 例如結果是” /usr/lib/libfuse.so.2” ，則通過`LD_LIBRARY_PATH=/usr/lib ossfs …`運行ossfs

#### 3. 我能忽略這個WARNING嗎?

- 最好不要，見這個bug
- Q: 為什麼用ossfs看到的檔案資訊（例如大小）與其他工具看到的不一致?

因為ossfs預設會快取檔案的元資訊（包括大小/許可權等），這樣就不需要每次ls的時候向OSS發送請求，加快速度。如果用戶通過其他程式（例如SDK/官網控制台/osscmd等）對檔案進行了修改，那麼有可能在ossfs中看到的檔案資訊沒有及時更新。

如果想禁止ossfs的緩存，那麼可以在掛載的時候加上如下參數：`-omax_stat_cache_size=0`

## 7 osscmd

---

### 7.1 快速安裝

#### 概述

osscmd是基於 Python 2.x 的命令列工具，支援Bucket管理、檔案管理等功能。



#### 说明:

非必要場景下建議使用`ossutil`代替`osscmd`。

#### · 使用場景

- API等級的開發、調試，比如發送特定格式的請求、分步驟執行分區上傳等。
- Bucket配置，不方便使用控制台情況下的Bucket配置，如logging/website/lifecycle等。

#### · 局限

- osscmd支援的運行環境包括Python 2.5/2.6/2.7，不支援Python 3.x。
- 在Python SDK 0.x基礎上開發，Python SDK 0.x已經不再維護，目前維護的`Python SDK`是2.x.x。
- OSSCMD不再支援新功能，只進行BUG修改，比如不支援低頻儲存/Archive Storage、跨區域複製、鏡像回源等。

強烈建議使用`ossutil`代替`osscmd`，`ossutil`具有下列優勢：

- `ossutil`支援Windows/Linux/Mac多種平台。
- `ossutil`基於Go SDK實現，安裝簡單、性能優越。
- `ossutil`命令簡單、幫助豐富、支援中文/英文雙語。

#### 環境要求

osscmd是隨Python sdk 0.x一起發布的，請點擊[這裡](#)下載。注意，Python SDK 2.x暫時並未提供相應版本的osscmd工具。

Python的版本要求要在2.5和2.7之間。SDK適用於Windows平台和Linux平台，但由於Python3.0並不完全相容2.x的版本，所以SDK暫時不支援3.0及以上的版本。

安裝好Python後：

- Linux shell環境下輸入python並回車，來查看Python的版本。如下所示：

```
Python 2.5.4 (r254:67916, Mar 10 2010, 22:43:17)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-46)] on linux2
Type "help", "copyright", "credits" or "license" for more informatio
n.
>>>
```

- Windows在cmd環境下輸入python並回車，來查看Python的版本。如下所示：

```
C:\Documents and Settings\Administrator>python
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (
Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more informatio
n.
>>>
```

以上說明python安裝成功。

- 異常情況，如Windows在cmd環境下輸入python並回車後，提示“不是內部或外部命令”，請檢查配置“環境變數” - “Path”，增加python的安裝路徑。如圖：

如果沒有安裝Python，可以從[python官網](#)獲取Python的安裝包。網站有詳細的安裝說明來指導用戶如何安裝和使用Python。

## 安裝使用

對下載的python SDK壓縮包進行解壓後，在osscmd所在目錄直接執行python osscmd + 操作即可。比如上傳一個檔案到bucket：

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

需要特別說明的是osscmd中用oss://bucket或者oss://bucket/object表示這是一個bucket或者object。oss://只是一種資源的表示方式，沒有其他意義。

如果需要詳細的命令列表輸入：python osscmd。

如果需要詳細的參數列表說明輸入：python osscmd help。

## 7.2 使用樣本

### 安裝配置osscmd

在Linux或者Windows上下載SDK安裝包後，解壓縮後就可以使用 osscmd了。

使用時直接調用python osscmd即可獲取相應的說明。每種命令有兩種執行模式。以查詢用戶所建立的bucket為例子。執行的是gs命令，get service的簡寫。

- 方法1: 不指定ID和KEY, osscmd從預設檔案中讀取ID和KEY。

```
$ python osscmd gs
can't get accessid/accesskey, setup use : config --id=accessid --key
=accesskey
```



说明:

如果出現這樣的提示表明沒有配置好ID和KEY, 見步驟2中提示的配置命令。

如果配置好ID和KEY, 並且ID和KEY有效, 執行

```
$ 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從命令列中讀取ID和KEY。如果ID和KEY有效, 執行後會得之後的結果。

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

如果要配置用戶的ID和KEY到預設的檔案中, 請運行如下命令用來配置訪問OSS所需要的ID和KEY。預設的OSS HOST為oss.aliyuncs.com。

```
$python osscmd config --id=your_id --key=your_key --host=your_endpo
int
```

如果出現類似“Your configuration is saved into”的提示表明ID和KEY已經保存成功了。

## 基礎操作

- 列出建立的bucket

```
$python osscmd getallbucket
```

如果是剛剛使用OSS的用戶因為沒有建立bucket, 輸出是空。

- 建立bucket

建立一個Bucket名字為mybucketname的bucket。

```
$python osscmd createbucket mybucketname
```

建立“mybucketname”的bucket, 有可能不成功。因為OSS中的bucket名字是全域唯一的, 並且有人已經建立了這個bucket。這個時候需要換一個名字。例如在bucket名字中加入特定的日期。

- 查看是否建立成功

```
$python osscmd getallbucket
```

如果沒有成功請檢查osscmd返回的錯誤資訊。

- 查看Object

成功建立bucket後，查看bucket中有哪些object。

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

由於bucket中還沒有object，輸出是空的。

- 上傳object

向bucket中上傳一個object。假如本地檔案名叫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 為 Linux 命令，Windows下無此命令。

- 再次查看Object

如果建立成功，再次查看bucket中有哪些object。

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

- 下載object

從bucket中下載object到本地檔案，並比對下載的檔案的md5值

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



说明:

md5sum 為 Linux 命令，Windows下無此命令。

- 刪除object

```
$ python osscmd delete oss://mybucketname/test_object
```

- 刪除bucket



说明:

如果bucket中還有object的話則這個bucket不能被刪除。

```
$ python osscmd deletebucket mybucketname
```

使用lifecycle

- 配置一個lifecycle的xml格式的文字檔

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

這個表示刪除bucket下，以log\_backup/ 為首碼的，並且相對目前時間超過2天的object。

詳細的規則配置可以參考[API文檔](#)

- 寫入lifecycle

```
python osscmd putlifecycle oss://mybucket lifecycle.xml
0.150(s) elapsed
```

- 讀取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
```

- 刪除lifecycle

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

- 讀取lifecycle

```
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!
```

## 防盜鏈設定

- 允許空referer訪問

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

- 獲取設定的referer

```
$osscmd getreferer oss://test
<?xml version="1.0" encoding="UTF-8"?>
<RefererConfiguration>
 <AllowEmptyReferer>true</AllowEmptyReferer>
 <RefererList />
</RefererConfiguration>
```

- 不允許空referer, 只允許referer為test的請求

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

- 獲取設定的referer

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

- 不允許空referer,只允許referer為test和test1的請求

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

- 獲取設定的referer

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

```
</RefererConfiguration>
```

使用logging

- 設定logging

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

- 獲取設定的logging

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

## 7.3 有關Bucket命令

config

**命令說明：**

```
config --id=[accessid] --key=[accesskey] --host=[host] --sts_token=[sts_token]
```

**使用示範：**

- `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)

**命令說明：**

```
getallbucket(gs)
```

用來顯示用戶建立的bucket。gs是get service的簡寫。(gs)表示和getallbucket是同樣的效果。

**使用示範：**

- `python osscmd getallbucket`
- `python osscmd gs`

createbucket(cb,mb,pb)

**命令說明：**

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

建立bucket的命令，cb是create bucket的簡寫，mb是make bucket的簡寫，pb是put bucket的簡寫，oss://bucket表示bucket。--acl參數可以傳入，也可以不傳入。這幾個命令都是同樣的效果。

使用示範：

- `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)

命令說明：

`deletebucket(db) oss://bucket`

刪除bucket的命令，db是delete bucket的簡寫。deletebucket和db是同樣的效果。

使用示範：

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

deletewholebucket

注意：該命令十分危險，將會刪除所有的數據，並且不可恢復。請慎重使用。

命令說明：

`deletewholebucket oss://bucket`

刪除bucket及其內部object以及multipart相關的內容。

使用示範：

- `python osscmd deletewholebucket oss://mybucket`

getacl

命令說明：

`getacl oss://bucket`

獲取bucket的存取控制許可權

使用示範：

- `python osscmd getacl oss://mybucket`

## setacl

## 命令說明：

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

修改bucket的存取控制許可權。acl只允許為private, public-read, public-read-write三個當中的一個。

## 使用示範：

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

## putlifecycle

## 命令說明：

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

設定lifecycle規則。其中lifecycle.xml為xml格式的lifecycle設定檔，詳細的規則配置可以參考[API文檔](#)

## 使用示範：

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

其中lifecycle.xml為XML格式的lifecycle配置規則，舉例為：

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

## getlifecycle

## 命令說明：

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

獲取該Bucket lifecycle規則。

## 使用示範：

- `python osscmd getlifecycle oss://mybucket`

## deletelifecycle

### 命令說明：

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

刪除該bucket下所有的lifecycle規則。

### 使用示範：

- `python osscmd deletelifecycle oss://mybucket`

## putreferer

### 命令說明：

```
osscmd putreferer oss://bucket --allow_empty_referer=[true|false]
--referer=[referer]
```

設定防盜鏈規則。其中參數`allow_empty_referer`用來設定是否允許為空，為必選參數。參數`referer`來設定允許訪問的白名單，比如“`www.test1.com,www.test2.com`”，以“`,`”作為分隔。詳細的配置規則參考[產品文檔](#)。

### 使用示範：

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

## getreferer

### 命令說明：

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

獲取該Bucket下防盜鏈設定規則。

### 使用示範：

- `python osscmd getreferer oss://mybucket`

## putlogging

### 命令說明：

```
osscmd putlogging oss://source_bucket oss://target_bucket/[prefix]
```

其中`source_bucket`表示需要記錄日誌的bucket，而`target_bucket`則是用來存放產生的日誌。同時可以對源bucket產生的記錄檔設定首碼，方便用戶歸類查詢。

### 使用示範：

- `python osscmd getlogging oss://mybucket`

getlogging

**命令說明：**

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

獲取該bucket的logging設定規則，返回為一個xml。

**使用示範：**

- `python osscmd getlogging oss://mybucket`

## 7.4 有關Object命令

ls(list)

**命令說明：**

```
ls(list) oss://bucket/[prefix] [marker] [delimiter] [maxkeys]
```

列出bucket中的object。

**使用示範：**

- `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`

**命令說明：**

```
ls(list) oss://bucket/[prefix] --marker=xxx --delimiter=xxx --maxkeys=xxx
--encoding_type=url
```

列出bucket中的object。其中encoding\_type可以指定傳輸中使用的編碼，當指定為url編碼時，能夠支援顯示含控制字元的object。

**使用示範：**

- `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

**命令說明：**

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

建立一個以“/”結尾的object，並且size為0。

**使用示範：**

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

## listallobject

**命令說明：**

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

顯示bucket下所有的object，可以指定prefix來顯示。

**使用示範：**

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

## deleteallobject

**命令說明：**

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

刪除bucket下所有的object，可以指定特定的prefix來刪除。

**使用示範：**

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

## downloadallobject

**命令說明：**

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

將bucket下的object下載到本地目錄，並且保持目錄結構。可以指定prefix下載。—replace=false表示如果下載時，本地已經存在同名檔案，不會覆蓋。true則會覆蓋。同時可以通過thread\_num來配置下載線程。

**使用示範：**

- `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

### 命令說明：

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

將bucket下的object下載到本地目錄，並且保持目錄結構。可以指定prefix下載。—replace=false表示如果下載時，本地已經存在同名檔案，不會覆蓋。true則會覆蓋。同downloadallobject 效果一樣。

### 使用示範：

- `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

### 命令說明：

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

將本地目錄裡的檔案上傳到bucket中。例如localdir為/tmp/

裡面有a/b, a/c, a三個檔案，則上傳到OSS中為oss://bucket/a/b, oss://bucket/a/c, oss://bucket/a。如果指定了prefix為mytest，則上傳到OSS中為oss://bucket/mytest/a/b, oss://bucket/mytest/a/c, oss://bucket/mytest/a。

--check\_point=check\_point\_file是指定檔案。指定檔案後，osscmd會將已經上傳的本地檔案以時間戳記的方式放到check\_point\_file中，uploadfromdir命令會將正在上傳的檔案的時間戳記和check\_point\_file記錄的時間戳記進行比較。如果有變化則會重新上傳，否則跳過。預設情況下是沒有check\_point\_file的。--replace=false表示如果下載時，本地已經存在同名檔案，不會覆蓋。true則會覆蓋。--check\_md5=false表示上傳檔案時，不會做攜帶Content-MD5要求標頭校驗。true則會做校驗。

注意：由於check\_point\_file檔案中記錄的是上傳的所有檔案的。所以當上傳檔案特別多的時候，check\_point\_file會特別巨大。

#### 使用示範：

- `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

##### 命令說明：

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

上傳一個本地的檔案到bucket中，可以指定object的content-type，或則指定自訂的headers。--check\_md5=false表示上傳檔案時，不會做攜帶Content-MD5要求標頭校驗。true則會做校驗。

##### 使用示範：

- `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

##### 命令說明：

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

將本地檔案以object group的形式上傳。不推薦使用。--check\_md5=false表示上傳檔案時，不會做攜帶Content-MD5要求標頭校驗。true則會做校驗。

**使用示範：**

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

get

**命令說明：**

```
get oss://bucket/object localfile
```

將object下載到本地檔案。

**使用示範：**

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

multiget(multi\_get)

**命令說明：**

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

將object以多線程的方式下載到本地檔案。同時可以配置線程數。

**使用示範：**

- `python osscmd multiget oss://mybucket/myobject /tmp/localfile`
- `python osscmd multi_get oss://mybucket/myobject /tmp/localfile`

cat

**命令說明：**

```
cat oss://bucket/object
```

讀取object的內容，直接列印出來。在object內容比較大的時候請不要使用。

**使用示範：**

- `python osscmd cat oss://mybucket/myobject`

meta

**命令說明：**

```
meta oss://bucket/object
```

讀取object的meta資訊，列印出來。meta資訊包括content-type，檔案長度，自訂meta等內容。

**使用示範：**

- `python osscmd meta oss://mybucket/myobject`

copy

**命令說明：**

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

將源bucket的源object 複製到目的bucket中的目的object。

**使用示範：**

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

rm(delete,del)

**命令說明：**

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

刪除object。當指定encoding-type為url編碼時，傳入待刪除的字串也需為url編碼。

**使用示範：**

- `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)

**命令說明：**

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

生成一個包含簽名的URL，並指定逾時的時間。適用於bucket為私有時將特定的object提供給他人訪問。

**使用示範：**

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

## 7.5 有關Multipart命令

init

**命令說明：**

```
init oss://bucket/object
```

初始化生成一個Upload ID。這個Upload ID可以配合後面的multiupload命令來使用。

**使用示範：**

```
python osscmd init oss://mybucket/myobject
```

listpart

**命令說明：**

```
listpart oss://bucket/object --upload_id=xxx
```

顯示指定object的Upload ID下已經上傳的Parts。相關概念見OSS API文檔。必須要指定Upload ID。

**使用示範：**

```
python osscmd listpart oss://mybucket/myobject --upload_id=
75835E389EA648C0B93571B6A46023F3
```

listparts

**命令說明：**

```
listparts oss://bucket
```

顯示bucket中未完成的multipart Upload ID和object。一般在刪除bucket提示bucket非空的情況下可以用這個命令查看是否有multipart相關的內容。

**使用示範：**

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

getallpartsize

**命令說明：**

```
getallpartsize oss://bucket
```

顯示bucket中還存在的Upload ID已經上傳的Parts的總大小。

**使用示範：**

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

cancel

#### 命令說明：

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

終止Upload ID對應的Multipart Upload事件。

#### 使用示範：

```
python osscmd cancel oss://mybucket/myobject --upload_id=
D9D278DB6F8845E9AFE797DD235DC576
```

multiupload(multi\_upload,mp)

#### 命令說明：

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

將本地檔案以multipart的方式上傳到OSS。

#### 使用示範：

- `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`

#### 命令說明：

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

將本地檔案以multipart的方式上傳到OSS。本地檔案劃分的塊數由max\_part\_num來指定。這個命令在實現的時候，會先去判斷Upload ID對應的Parts的ETag是否和本地檔案的MD5值是否相等，相等則跳過上傳。所以如果在使用之前生成一個Upload ID，作為參數傳進來。即使上傳沒有成功，重複執行相同的multiupload命令可以達到一個斷點續傳的效果。--check\_md5=false表示上傳檔案時，不會做攜帶Content-MD5要求標頭校驗。true則會做校驗。

#### 使用示範：

- `python osscmd multiupload /tmp/localfile.txt oss://mybucket/object
--upload_id=
D9D278DB6F8845E9AFE797DD235DC576`
- `python osscmd multiup_load /tmp/localfile.txt oss://mybucket/object`

```
--thread_num=5
```

- `python osscmd mp /tmp/localfile.txt oss://mybucket/object --max_part_num=100`

## copylargefile

### 命令說明：

```
copylargefile oss://source_bucket/source_object oss://target_bucket/target_object
--part_size=10*1024*1024 --upload_id=xxx
```

對於超過1G的大檔案進行複製時，採用multipart的方式將object複製到指定位置，（源bucket必須與目標bucket處於同一region）。其中upload\_id為選擇性參數，如果當需要對某一次multipart copy事件進行續傳的時候，可以傳入該事件的upload\_id。part\_size用來設定分塊大小，分塊最小需要大於100KB，最多支援10000塊分塊。如果part\_size設定值導致與OSS限制衝突，程式會幫你自動調節分塊大小。

### 使用示範：

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

## uploadpartfromfile (upff)

### 命令說明：

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

主要用於測試，不推薦使用。

## uploadpartfromstring(upfs)

### 命令說明：

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

主要用於測試，不推薦使用。