

Alibaba Cloud ApsaraVideo for Media Processing

SDK Reference

Issue: 20200604









Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

- 1.** You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
- 2.** No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
- 3.** The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
- 4.** This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

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

Document conventions

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

Style	Description	Example
{ } or {a b}	This format is used for a required value, where only one item can be selected.	switch {active stand}

Contents

Legal disclaimer.....	1
Document conventions.....	1
1 Create AccessKey.....	1
2 Upload SDK.....	2
2.1 Instructions for use.....	2
2.2 JavaScript version.....	5
2.3 Android version.....	7
2.4 iOS version.....	10
2.5 Upload SDK.....	14
3 Transcoding SDKs.....	17
3.1 SDK introduction.....	17
3.2 Multi-Region support.....	17
3.3 Java SDK.....	17
3.3.1 Overview.....	17
3.3.2 Installation.....	18
3.3.3 Quick start.....	19
3.3.4 Transcoding.....	21
3.3.5 Watermarks.....	24
3.3.6 Screenshot.....	27
3.3.7 HLS data encryption.....	30
3.3.8 Create HLS standard encryption workflow.....	32
3.3.9 MPS queues management.....	34
3.3.10 Query media - Use OSS file address.....	36
3.3.11 Add media.....	37
3.3.12 Splicing and simple cutting.....	38
3.3.13 Add media workflow.....	41
3.3.14 Splicing-opening and ending scenes.....	44
3.4 Python SDK.....	48
3.4.1 Overview.....	48
3.4.2 Installation.....	48
3.4.3 Quick start.....	48
3.4.4 Transcoding.....	49
3.4.5 Watermarks.....	51
3.4.6 Screenshot.....	54
3.4.7 Query media - Use OSS file address.....	56
3.4.8 Create HLS standard encryption workflow.....	57
3.4.9 Splicing and simple cutting.....	59
3.4.10 Splicing-opening and ending scenes.....	61
3.4.11 Add media.....	63
3.4.12 Add media workflow.....	63

3.5 PHP SDK.....	65
3.5.1 Overview.....	65
3.5.2 Installation.....	65
3.5.3 Quick start.....	65
3.5.4 Transcoding.....	66
3.5.5 Watermarks.....	69
3.5.6 Screenshot.....	72
3.5.7 Splicing and simple cutting.....	74
3.5.8 Splicing-opening and ending scenes.....	76
3.5.9 Create HLS standard encryption workflow.....	78
3.5.10 MPS queue management.....	81
3.5.11 Query media - Use OSS file address.....	82
3.5.12 Add media.....	83
3.5.13 Add media workflow.....	83
4 Demo project.....	86

1 Create AccessKey

Perform the following steps to create AccessKey.

1. Log on to the Alibaba Cloud website.
2. Visit the [AccessKey console](#).
3. Create and manage your AccessKey.

2 Upload SDK

2.1 Instructions for use

The upload SDK provides the file list management and upload control functions. File list management allows you to add, delete, cancel, resume, retrieve, and clear files. Upload control allows you to start, stop, pause, and resume file upload. The SDK provides callback events to monitor the status and progress changes during upload.

Upload process

Initialize the upload SDK > Select a file > Add the file to the list > Start upload > Upload completion events.

- Initialize the upload SDK

The AccessKey and token can be used to grant permissions during initialization. Considering the security of the AccessKey stored on the client, it is recommended that the AccessKey be used only for testing and the token be used in the production environment. For more information, see [Developer guide > Upload a video file > Overview](#).

- Select a file

Select a local file to be uploaded.

- Add the file to the list

Add all files to be uploaded to the list using `addFile`.

- Start upload.

Call `start` to start the upload process.

- Upload completion events

The events include `OnUploadSucceed` and `OnUploadFailed`.

Concepts and descriptions

- Multipart upload and status

The SDK uses the multipart upload mode, in which the status is valid only for one execution. If the app exits due to a specific reason (for example, shutdown, closing the browser page, closing the app, or abnormal app exit), the file must be uploaded again.

- Authorization expiration

The token is only valid for a certain period. After the token expires, the upload process is interrupted and cannot be automatically resumed. To resume upload, obtain a new token from the backend and call the `resumeUploadWithAuth` function.

- Switching between a 3G/4G network and a Wi-Fi network on the mobile end

To avoid traffic waste on 3G/4G networks, when the app switches to a 3G/4G network, it must automatically detect the network and call `pause` to suspend upload. After the app detects that a Wi-Fi network is used, it calls `resume` to resume upload.

- The SDKs of the following three types of terminals are provided:

- [HTML5](#): It can be integrated to PC browsers. The development language is JavaScript.
- [iOS](#): It can be integrated to iOS apps in the language of Object-C.
- [Android](#): It can be integrated to Android apps in the language of Java.

Features

File list management

API	Description
addFile	Adds a file to the list. Files are uploaded in order of addition.
deleteFile	Deletes a file from the list.
cancelFile	Cancels a file in the list but does not delete the file from the list. After this API is used, upload of this file is skipped (not supported by the JavaScript version).
resumeFile	Resumes the status of the file that is canceled in the list (not supported by the JavaScript version). This API is not used to resume upload.
listFiles	Obtains the list.
clearFiles	Clears the list. After this API is used, upload stops and files, even the files being uploaded, are cleared (not supported by the JavaScript version).

Upload control

API	Description
start	Starts upload.
stop	Stop uploading
pause	Stops upload (JavaScript version is not supported).
resume	Pauses upload (not supported by the JavaScript version).
resumeUploadWithToken	Uses a new token to resume upload after the existing token expires.

Callback events

Event	Description
OnUploadStarted	This event is triggered when upload of each file starts.
OnUploadSucceed	Upload succeeds.
OnUploadFailed	Upload fails. If resumable errors occur, for example the network has an exception or times out, the file upload can be resumed. If non-resumable errors occur, for example the upload credential is incorrect or the file does not exist, the file upload fails.
OnUploadProgress	The upload progress report. This event is triggered when multipart upload succeeds.
OnUploadTokenExpired	The token times out. To resume upload, obtain a new token from the server and call the <code>resumeUploadWithToken</code> function.
OnUploadRetry	This event is triggered when the status switches from normal to abnormal during upload. For example, the network has an exception or times out (not supported by the JavaScript version).
OnUploadRetryResume	This event is triggered when the status is resumed from abnormal to normal during upload (not supported by the JavaScript version).

2.2 JavaScript version

Installation

[Download the upload SDK](#)

Introduce the following two JavaScript scripts on the page:

```
<script src="aliyun-sdk.min.js"></script>
<script src="vod-sdk-upload-1.0.6.min.js"></script>
```

Initialization: create a VODUpload instance.

Set the callback function.

```
var uploader = new VODUpload({
  // Starts upload
  'onUploadstarted': function (uploadInfo) {
    log("onUploadStarted:" + uploadInfo.file.name + ", endpoint:" + uploadInfo.endpoint
    + ", bucket:" + uploadInfo.bucket + ", object:" + uploadInfo.object);
  }
  //File upload succeeds
  'onUploadSucceed': function (uploadInfo) {
    log("onUploadSucceed: " + uploadInfo.file.name + ", endpoint:" + uploadInfo.
    endpoint + ", bucket:" + uploadInfo.bucket + ", object:" + uploadInfo.object);
  },
  //File upload fails
  'onUploadFailed': function (uploadInfo, code, message) {
    log("onUploadFailed: file:" + uploadInfo.file.name + ",code:" + code + ", message:" +
    message);
  },
  // The file upload progress, in bytes
  'onUploadProgress': function (uploadInfo, totalSize, uploadedSize) {
    log("onUploadProgress:file:" + uploadInfo.file.name + ", fileSize:" + totalSize + ",
    percent:" + Math.ceil(uploadedSize * 100 / totalSize) + "%");
  },
  // Token expires
  'onUploadTokenExpired': function () {
    console.log("onUploadTokenExpired");
    // uploader.resumeUploadWithToken(accessKeyId, accessKeySecret, secretToken,
    expireTime);
  }
});
```

List management

- Add a file to be uploaded



Note:

The file size cannot exceed 10 GB.

Use the standard input mode for selecting a file.

```
<form action="">
<input type="file" name="file" id="files" multiple/>
```

```

</form>
userData = "";
document.getElementById("files")
  .addEventListener('change', function (event) {
    for(var i=0; i<event.target.files.length; i++) {
      // The logic code
    }
  });

```

Obtain the selected file and add it to the upload list.

```
uploader.addFile(event.target.files[i], endpoint, bucket, object, userData);
```



Note:

- event.target.files[i]: file list that user select,
- endpoint: endpoint of OSS,
- bucket: bucket of OSS,
- object: object of OSS,
- userData: the last parameter of addFile, a json object.

During uploading, obtain the attributes (the title, tag, description, category, and custom data) of a media set in the following way: The last parameter userData of the addFile function is a JSON object. The first-level VOD is required, and VOD contains the five attributes. Example:

```

var userData = '{"Vod":{"Title":"I am the title",
  "Description":"I am the description",
  "Cateoid": "1 ",
  "Tags":"tag1,tag2,tag3",
  "UserData":"user data"}}';

```

- Delete the uploaded file. `index` corresponds to the index of the elements in the list returned by `listFiles`.

```
uploader.deleteFile(index);
```

- Cancel upload of a single file.

```
uploader.cancelFile(index);
```

- Resume upload of a single file.

```
uploader.resumeFile(index);
```

- Obtain the upload file list.

```

uploader.listFiles();
var list = uploader.listFiles();
for (var i=0; i<list.length; i++) {

```



```
log("file:" + list[i].file.name + ", status:" + list[i].state + ", endpoint:" + list[i].  
endpoint + ", bucket:" + list[i].bucket + ", object:" + list[i].object);  
}
```

- Clear the upload file list.

```
uploader.cleanList();
```

Upload control

- Start upload.

```
uploader.startUpload();
```

- Stop upload

```
uploader.stopUpload();
```

- Resume upload after the upload credential becomes invalid.

```
uploader.resumeUploadWithToken(accessKeyId, accessKeySecret, secretToken,  
expireTime);
```

2.3 Android version

Environment requirements

Android 2.3 or a later version

Installation

[OSS Android SDK](#)

[Upload SDK](#)

- Directly introduce the JAR package

After you download the ZIP package of the VODUpload Android SDK, perform the following steps (applicable to Android Studio or Eclipse):

- Decompress the SDK to obtain the following JAR packages in the libs directory: aliyun-oss-sdk-android-xxx.jar, okhttp-2.7.0.jar, okio-2.6.0.jar, and aliyun-vod-upload-android-sdk-xxx.jar.
- Import the four JAR packages to the libs directory of the project.
- Set permissions

The following are the Android permissions required by the VODUpload Android SDK. Make sure that these permissions are already set in your AndroidManifest.xml file. Otherwise, the SDK cannot work normally.

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
"></uses-permission>
```

Create a VODUpload instance

Set the callback function.

```
VODUploadCallback callback = new VODUploadCallback() {
    /**
     * Triggered when the file upload starts
     */
    void onUploadStarted() {}
    /**
     * Callback after successful upload
     */
    void onUploadSucceed(UploadFileInfo info) {}
    /**
     * Upload failed
     */
    void onUploadFailed(UploadFileInfo info, String code, String message) {}
    /**
     * Callback upload progress
     * @param uploadedSize The number of uploaded bytes
     * @param totalSize The total number of required bytes
     */
    void onUploadProgress(UploadFileInfo info, long uploadedSize, long totalSize) {}
    /**
     * This API is called back after the upload credential expires
     * Obtain a new upload credential in the callback and call resumeUploadWithAuth to
     continue upload
     */
    void onUploadTokenExpired() {}
    /**
     * Triggered when the status switches from normal to abnormal during upload
```

```

*/
void onUploadRetry(String code, String message) {}
/**
 * Triggered when the status is resumed from abnormal during upload
 */
void onUploadRetryResume() {}
};
VODUploadClient uploader = new VODUploadClientImpl(getContext());

```

Initialization

Enter the authorization information in either of the following ways:

- AccessKey

It is simple but not safe. It is recommended that this way be used in the test environment.

```
uploader.init("<accessKeyId>", "<accessKeySecret>", callback);
```

- Token

It is safe but complex. It is recommended that this way be used in the production environment. A token is temporary and valid for a period. Therefore, sending a token is safe.

```
uploader.init("<accessKeyId>", "<accessKeySecret>", "<secretToken>", "<expireTime>",
callback);
```

List management

- Add a file to be uploaded.



Note:

The file size cannot exceed 4 GB.

```

uploader.addFile("<uploadFilePath>",
                "<endpoint>", // For example, the Hangzhou regio "http://oss-cn-hangzhou.
                aliyuncs.com"
                "<bucketName>", // Enter the actual bucket name
                "<objectKey>");

```

During uploading, obtain the attributes (the title, tag, description, category, cover URL, and custom data) of a media set in the following way: addFile contains a reload function, in which the last parameter is a VodInfo object. The definitions are as follows:

```

private String title;
private String desc;
private Integer catelId;
private List<String> tags;
private String userData;

```

```
private String coverUrl;
```

- Delete the uploaded file. `index` corresponds to the index of the elements in the list returned by `listFiles`.

```
uploader.deleteFile(index);
```

- Cancel upload of a single file in the list.

```
uploader.cancelFile(index);
```

- Resume upload of a single file in the list.

```
uploader.resumeFile(index);
```

- Obtain the upload file list.

```
List list = uploader.listFiles();
```

- Clear the upload file list.

```
upload.clearFiles();
```

Upload control

- Start upload.

```
uploader.start();
```

- Stop upload.

```
uploader.stop();
```

- Pause upload.

```
uploader.pause();
```

- Resume upload.

```
uploader.resume();
```

- Resume upload after the token is invalid.

```
uploader.resumeWithToken("<accessKeyId>", "<accessKeySecret>", "<secretToken>", "<expireTime>");
```

2.4 iOS version

Environment requirements

iOS 7.0 or a later version

Installation

OSS iOS SDK

Download the upload SDK

- Directly introduce the frameworks

Introduce the frameworks of the OSS iOS SDK and VODUpload iOS SDK.

In Xcode, drag the frameworks and drop them to your target, and select “Copy items if needed” in the displayed dialog box.

- Introduce the header file to your project

```
#import <VODUpload/VODUploadClient.h>
```



Note:

After you introduce the frameworks, add `-ObjC` to Other Linker Flags of Build Settings in your project. If the `-force_load` option has been configured for your project, add `-force_load <framework path>/AliyunOSSiOS`.

- Compatible with IPv6-Only networks.

The OSS mobile SDK has introduced the HTTPDNS for domain name resolution to avoid domain name resolution hijacking in a wireless network and directly uses IP addresses for requests to the server. In an IPv6-Only network, compatibility issues may occur. The app officially issued the review requirements for apps, requiring apps to be IPv6-only network compatible. To this end, the SDK starts to be compatible from V2.5.0. In the new version, apart from `-ObjC` settings, two system libraries must be introduced:

```
libresolv.tbd  
SystemConfiguration.framework
```

Create a VODUpload instance

Set the callback function.

```
OnUploadStartedListener testUploadStartedCallbackFunc = ^(UploadFileInfo* fileInfo) {  
    NSLog(@"upload started.");  
};  
OnUploadSucceedListener testSuccessCallbackFunc = ^(NSString* filePath){  
    NSLog(@"file:%@ upload success!", filePath);  
};  
OnUploadFailedListener testFailedCallbackFunc = ^(NSString* filePath, NSString* code,  
NSString* message){  
    NSLog(@"failed code = %@, error message = %@", code, message);  
};  
// Unit: byte
```

```

OnUploadProgressListener testProgressCallbackFunc = ^(NSString* filePath, long
uploadedSize, long totalSize) {
    NSLog(@"progress uploadedSize : %li, totalSize : %li", uploadedSize, totalSize);
};
OnUploadTokenExpiredListener testTokenExpiredCallbackFunc = ^{
    NSLog(@"token expired.");
    // get token and call resumeUploadWithAuth.
};
OnUploadRertyListener testUploadRertyListener = ^{
    NSLog(@"retry begin.");
};
OnUploadRertyResumeListener testUploadRertyResumeListener = ^{
    NSLog(@"retry resume.");
};
VODUploadListener *listener;
listener = [[VODUploadListener alloc] init];
listener.started = testUploadStartedCallbackFunc;
listener.success = testSuccessCallbackFunc;
listener.failure = testFailedCallbackFunc;
listener.progress = testProgressCallbackFunc;
listener.expire = testTokenExpiredCallbackFunc;
listener.retry = testUploadRertyListener;
listener.retryResume = testUploadRertyResumeListener;

```

Initialize the upload SDK

Enter the authorization information in either of the following ways:

- AccessKey

It is simple but not safe. It is recommended that this way be used in the test environment.

```

VODUploadClient *uploader;
[uploader init:<accessKeyId>
    accessKeySecret:<accessKeySecret>
    listener:listener];

```

- Token

It is safe but complex. It is recommended that this way be used in the production environment. A token is temporary and valid for a period. Therefore, sending a token is safe.

```

VODUploadClient *uploader;
[uploader init:<accessKeyId>
    accessKeySecret:<accessKeySecret>
    secretToken:<secretToken>
    expireTime:<expireTime>
    listener:listener];

```

- List Management

- Add a file to be uploaded.



Note:

The file size cannot exceed 4 GB.

```
[uploader addFile:<uploadFilePath>
  endpoint:<endpoint> //Example: 'http://oss-cn-hangzhou.aliyuncs.com'
  bucket:<bucketName> //Enter the actual bucket name
  object:<objectKey>];
```

During uploading, obtain the attributes (the title, tag, description, category, cover URL, and custom data) of a media set in the following way: addFile contains a reload function, in which the last parameter is a VodInfo object. The definitions are as follows:

```
@interface VodInfo : NSObject
@property (nonatomic, strong) NSString* title;
@property (nonatomic, strong) NSString* tags;
@property (nonatomic, strong) NSString* desc;
@property (nonatomic, strong) NSNumber* catelId;
@property (nonatomic, strong) NSString* userData;
```

```
@property (nonatomic, strong) NSString* coverUrl;
```

- Delete the uploaded file.

```
[uploader deleteFile:<index>];
```

- Cancel upload of a single file in the list.

```
[uploader cancelFile:<index>];
```

- Resume upload of a single file in the list.

```
[uploader resumeFile:<index>];
```

- Obtain the upload file list.

```
[uploader listFiles];
```

- Clear the upload file list.

```
[uploader clearFiles];
```

Upload control

- Start upload.

```
[uploader start];
```

- Stop upload.

```
[uploader stop];
```

- Pause upload.

```
[uploader pause];
```

- Resume upload.

```
[uploader resume];
```

- Resume upload after the token is invalid.

```
[uploader resumeWithToken:<accessKeyId>  
accessKeySecret:<accessKeySecret>  
secretToken:<secretToken>  
expireTime:<expireTime>]
```

2.5 Upload SDK

Latest versions

- [SDK for JavaScript 1.0.6, sample code](#)
- [SDK for iOS 1.0.7, sample code](#)

- [SDK for Android 1.0.6, sample code](#)

1.0.7

- [SDK for iOS 1.0.7](#)

Updates

- Added support for bitcode

1.0.6

- [SDK for JavaScript 1.0., sample code](#)
- [SDK for iOS 1.0.6, sample code](#)
- [SDK for Android 1.0.6, sample code](#)

Updates:

- For Android and iOS, index serial numbers, instead of file names, are used in the parameters of list management APIs (cancelFile, deleteFile, and resumeFile) to correspond to listFiles.
- Both the on-demand and traditional OSS upload modes are supported.
- The on-demand upload mode is added in the sample code.

1.0.5

- [SDK for JavaScript 1.0.5, sample code](#)
- [SDK for iOS 1.0.5, sample code](#)
- [SDK for Android 1.0.5, sample code](#)

Updates

- The pause, resuming, and network exceptions are added.
- The title, tag, cover URL, category ID, description, and custom data of the media set can be specified for a video to be uploaded.
- The SDK for iOS integrates OSS 2.6.0 and supports the App Transport Security (ATS) standard of Apple Inc.

0.0.4

- [SDK for JavaScript 0.0., sample code](#)
- [SDK for iOS 0.0.4, sample code](#)
- [SDK for Android 0.0.4, sample code](#)

Updates

- First release

3 Transcoding SDKs

3.1 SDK introduction

SDK introduction details the installation, application, and example codes for MPS SDK.

MPS SDK is based on Alibaba Cloud SDK. For more information about Alibaba Cloud SDK, see:

- [Alibaba Cloud SDK](#)
- [Alibaba Cloud API](#)

3.2 Multi-Region support

MPS is available in many regions. For more information about regions list and RegionId corresponding to SDK, see:

- [MPS service regions](#)
- [RegionId corresponding to regions](#)

3.3 Java SDK

3.3.1 Overview

MPS Java SDK is based on Alibaba Cloud Java SDK. This article introduces the basic knowledge about Alibaba Cloud Java SDK.

- [Alibaba Cloud SDK Developer Guide > Java SDK > Getting started](#)
- [Alibaba Cloud SDK Developer Guide > Java SDK > Use Java SDK](#)
- [Alibaba Cloud Java SDK GitHub](#)

After learning about the basic knowledge, you can perform MPS Java SDK installation. For more information, see [MPS > SDK Reference > Java SDK > Installation](#).

3.3.2 Installation

This article introduces Maven installation method recommended by Alibaba Cloud Java SDK. Specifically, it includes two steps. First, add the Maven repository of Alibaba Cloud Java SDK to the pom.xml configuration file. Then, add MPS dependency.

1. Add a Maven repository.

```
<repositories>
  <repository>
    <id>sonatype-nexus-staging</id>
    <name>Sonatype Nexus Staging</name>
    <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>
```

2. Add dependency.

Alibaba Java SDK Core and MPS Java SDK detailed version:

- [Alibaba Cloud Java SDK Core](#)
- [MPS Java SDK](#)

Take the core of V3.5.0 SDK Core and V2.5.2 MPS SDK as an example:

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>3.5.0</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-mts</artifactId>
  <version>2.5.2</version>
</dependency>
```

In addition, a json repository dependency optional. In MPS API, many parameters are defined through json. And many json repositories for java are available, you can select the repository that you are familiar with. Take V1.2.46 [fastjson](#) as an example:

```
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.46</version>
</dependency>
```

pom.xml example:

```
<? xml version="1.0" encoding="UTF-8"? >
```

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org
/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>mps-demo-project</groupId>
  <artifactId>mps-demo-project</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <repositories>
    <repository>
      <id>sonatype-nexus-staging</id>
      <name>Sonatype Nexus Staging</name>
      <url>https://oss.sonatype.org/service/local/staging/deploy/maven2/</url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>true</enabled>
      </snapshots>
    </repository>
  </repositories>
  <dependencies>
    <dependency>
      <groupId>com.aliyun</groupId>
      <artifactId>aliyun-java-sdk-core</artifactId>
      <version>3.5.0</version>
    </dependency>
    <dependency>
      <groupId>com.aliyun</groupId>
      <artifactId>aliyun-java-sdk-mts</artifactId>
      <version>2.5.2</version>
    </dependency>
    <dependency>
      <groupId>com.alibaba</groupId>
      <artifactId>fastjson</artifactId>
      <version>1.2.46</version>
    </dependency>
  </dependencies>
  <build>
    <finalName>${artifactId}-${version}</finalName>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>2.3.2</version>
        <configuration>
          <source>1.6</source>
          <target>1.6</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>

```

3.3.3 Quick start

This article describes the Java SDK quick start process.

1. Create AcsClient instance.

```

DefaultProfile profile = DefaultProfile.getProfile(
    mpsRegionId, // region ID

```

```

        accessKeyId, //AccessKey ID of RAM account
        accessKeySecret); //Access Key Secret of RAM account
IAcsClient client = new DefaultAcsClient(profile);

```

2. Create request, and set parameters.

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. Initiate API request, and display the return value.

```

response = client.getAcsResponse(request);
System.out.println("PipelineName is:" + response.getPipelineList().get(0).getName());
System.out.println("PipelineId is:" + response.getPipelineList().get(0).getId());

```

Full code

```

import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class Quick {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String[] mpsRegionIds = new String[] {
        "cn-hangzhou", "cn-beijing", "cn-shenzhen", "cn-shanghai",
        "cn-hongkong", "us-west-1", "ap-southeast-1", "ap-northeast-1",
        "eu-central-1", "ap-south-1"
    };
};
public static void main(String[] args) {
    for (String mpsRegionId : mpsRegionIds) {
        System.out.println("region id is:" + mpsRegionId);
        // Create DefaultAcsClient instance and finish initialization
        DefaultProfile profile = DefaultProfile.getProfile(
            mpsRegionId, // region ID
            accessKeyId, // AccessKey ID of RAM account
            accessKeySecret); // Access Key Secret of RAM account
        IAcsClient client = new DefaultAcsClient(profile);
        // Create API request and set parameters
        SearchPipelineRequest request = new SearchPipelineRequest();
        //initiate request and handle response or exceptions
        SearchPipelineResponse response;
        try {
            response = client.getAcsResponse(request);
            System.out.println("PipelineName is:" + response.getPipelineList().get(0).
getName());
            System.out.println("PipelineId is:" + response.getPipelineList().get(0).getId());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
}

```

```
}
```

3.3.4 Transcoding

1. Create AcsClient instance.

```
DefaultProfile profile = DefaultProfile.getProfile(  
    mpsRegionId, // region ID  
    accessKeyId, // AccessKey ID of RAM account  
    accessKeySecret); // Access Key Secret of RAM account  
IAcsClient client = new DefaultAcsClient(profile);
```

2. Create request, and set parameters.

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. Transcoding parameters.

- Input

```
JSONObject input = new JSONObject();  
input.put("Location", ossLocation);  
input.put("Bucket", ossBucket);  
input.put("Object", URLEncoder.encode(ossInputObject, "utf-8"));  
request.setInput(input.toJSONString());
```

- Output

```
String outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8");  
JSONObject output = new JSONObject();  
output.put("OutputObject", outputOSSObject);
```

- Container

```
JSONObject container = new JSONObject();  
container.put("Format", "mp4");  
output.put("Container", container.toJSONString());
```

- Video

```
JSONObject video = new JSONObject();  
video.put("Codec", "H. 264");  
video.put("Bitrate", "1500");  
video.put("Width", "1280");  
video.put("Fps", "25");  
output.put("Video", video.toJSONString());
```

- Audio

```
JSONObject audio = new JSONObject();  
audio.put("Codec", "AAC");  
audio.put("Bitrate", "128");  
audio.put("Channels", "2");  
audio.put("Samplerate", "44100");
```

```
output.put("Audio", audio.toJSONString());
```

- TemplateId

```
output.put("TemplateId", templateId);
```

- PipelineId

```
request.setPipelineId(pipelineId);
```

4. Initiate API request and display returned value.

```
SubmitJobsResponse response;
response = client.getAcsResponse(request);
System.out.println("RequestId is:" + response.getRequestId());
if (response.getJobResultList().get(0).getSuccess()) {
    System.out.println("JobId is:" + response.getJobResultList().get(0).getJob().getJobId());
} else {
    System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0).
        getCode() +
        " message:" + response.getJobResultList().get(0).getMessage());
}
```

Full codes

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class SimpleTranscode {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String templateId = "S00000001-200010";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "xxx";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output.mp4";
    public static void main(String[] args) {
        // Create DefaultAcsClient instance and complete initialization
        DefaultProfile profile = DefaultProfile.getProfile(
            mpsRegionId, // region D
            accessKeyId, //AccessKey ID of RAM account
            accessKeySecret); // Access Key Secret of RAM account
        IAcsClient client = new DefaultAcsClient(profile);
        // Create API request and set parameters
        SubmitJobsRequest request = new SubmitJobsRequest();
        // Input
        JSONObject input = new JSONObject();
        input.put("Location", ossLocation);
        input.put("Bucket", ossBucket);
        try {
            input.put("Object", URLEncoder.encode(ossInputObject, "utf-8"));
        } catch (UnsupportedEncodingException e) {
```



```

        throw new RuntimeException("input URL encode failed");
    }
    request.setInput(input.toJSONString());
    // Output
    String outputOSSObject;
    try {
        outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8");
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("output URL encode failed");
    }
    JSONObject output = new JSONObject();
    output.put("OutputObject", outputOSSObject);
    // Ouput->Container
    JSONObject container = new JSONObject();
    container.put("Format", "mp4");
    output.put("Container", container.toJSONString());
    // Ouput->Video
    JSONObject video = new JSONObject();
    video.put("Codec", "H. 264");
    video.put("Bitrate", "1500");
    video.put("Width", "1280");
    video.put("Fps", "25");
    output.put("Video", video.toJSONString());
    // Ouput->Audio
    JSONObject audio = new JSONObject();
    audio.put("Codec", "AAC");
    audio.put("Bitrate", "128");
    audio.put("Channels", "2");
    audio.put("Samplerate", "44100");
    output.put("Audio", audio.toJSONString());
    // Ouput->TemplateId
    output.put("TemplateId", templateId);
    JSONArray outputs = new JSONArray();
    outputs.add(output);
    request.setOutputs(outputs.toJSONString());
    request.setOutputBucket(ossBucket);
    request.setOutputLocation(ossLocation);
    // PipelineId
    request.setPipelineId(pipelineId);
    // Initiate request and handle response or exceptions
    SubmitJobsResponse response;
    try {
        response = client.getAcsResponse(request);
        System.out.println("RequestId is:" + response.getRequestId());
        if (response.getJobResultList().get(0).getSuccess()) {
            System.out.println("JobId is:" + response.getJobResultList().get(0).getJob().
getJobId());
        } else {
            System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0)
).getCode() +
                " message:" + response.getJobResultList().get(0).getMessage());
        }
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
}

```

```
}
```

3.3.5 Watermarks

1. Create AcsClient instance.

```
DefaultProfile profile = DefaultProfile.getProfile(
    mpsRegionId, // Region ID
    accessKeyId, // AccessKey ID
    accessKeySecret); // Access Key Secret
IAcsClient client = new DefaultAcsClient(profile);
```

2. Create request, and set parameters.

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. Set transcoding parameters.

- Image watermark

```
// Image Watermark
JSONObject imageWatermarkInput = new JSONObject();
imageWatermarkInput.put("Location", ossLocation);
imageWatermarkInput.put("Bucket", ossBucket);
try {
    imageWatermarkInput.put("Object", URLEncoder.encode(imageWatermarkObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("imageWatermark Input URL encode failed");
}
JSONObject imageWatermark = new JSONObject();
imageWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
imageWatermark.put("Type", "Image");
imageWatermark.put("InputFile", imageWatermarkInput);
imageWatermark.put("ReferPos", "TopRight");
imageWatermark.put("Width", "0.05");
imageWatermark.put("Dx", "0");
imageWatermark.put("Dy", "0");
```

- Text watermark

```
// Text Watermark
JSONObject textConfig = new JSONObject();
textConfig.put("Content", "5rWL6K+V5paH5a2X5rC05Y2w");
textConfig.put("FontName", "SimSun");
textConfig.put("FontSize", "16");
textConfig.put("FontColor", "Red");
textConfig.put("FontAlpha", "0.5");
textConfig.put("Top", "10");
textConfig.put("Left", "10");
JSONObject textWatermark = new JSONObject();
textWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
textWatermark.put("Type", "Text");
textWatermark.put("TextWaterMark", textConfig.toJSONString());
```

- Video watermark

```
// Video Watermark
JSONObject videoWatermarkInput = new JSONObject();
```

```

videoWatermarkInput.put("Location", ossLocation);
videoWatermarkInput.put("Bucket", ossBucket);
try {
    videoWatermarkInput.put("Object", URLEncoder.encode(videoWatermarkObject
, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("videoWateetmark Input URL encode failed");
}
JSONObject videoWatermark = new JSONObject();
videoWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
videoWatermark.put("Type", "Image");
videoWatermark.put("InputFile", videoWatermarkInput);
videoWatermark.put("ReferPos", "BottomLeft");
videoWatermark.put("Height", "240");
videoWatermark.put("Dx", "0");
videoWatermark.put("Dy", "0");

```

4. Initiate API request and display returned value.

```

SubmitJobsResponse response;
response = client.getAcsResponse(request);
System.out.println("RequestId is:" + response.getRequestId());
if (response.getJobResultList().get(0).getSuccess()) {
    System.out.println("JobId is:" + response.getJobResultList().get(0).getJob().getJobId
());
} else {
    System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0).
getCode() +
        " message:" + response.getJobResultList().get(0).getMessage());
}

```

Full codes

```

package com.aliyun.mts;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class Watermark {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String watermarkTemplateId = "xxx";
    private static String templateId = "S00000001-200030";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "presigned";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output.mp4";
    private static String imageWatermarkObject = "logo.png";
    private static String videoWatermarkObject = "logo.mov";
    public static void main(String[] args) {
        // DefaultAcsClient
        DefaultProfile profile = DefaultProfile.getProfile(
            mpsRegionId, // Region ID
            accessKeyId, // AccessKey ID

```

```

        accessKeySecret); // Access Key Secret
IAcsClient client = new DefaultAcsClient(profile);
// request
SubmitJobsRequest request = new SubmitJobsRequest();
// Input
JSONObject input = new JSONObject();
input.put("Location", ossLocation);
input.put("Bucket", ossBucket);
try {
    input.put("Object", URLEncoder.encode(ossInputObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("input URL encode failed");
}
request.setInput(input.toJSONString());
// Output
String outputOSSObject;
try {
    outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8");
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("output URL encode failed");
}
JSONObject output = new JSONObject();
output.put("OutputObject", outputOSSObject);
// Ouput->TemplateId
output.put("TemplateId", templateId);
// Image Watermark
JSONObject imageWatermarkInput = new JSONObject();
imageWatermarkInput.put("Location", ossLocation);
imageWatermarkInput.put("Bucket", ossBucket);
try {
    imageWatermarkInput.put("Object", URLEncoder.encode(imageWatermarkObject,
"utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("imageWatetmark Input URL encode failed");
}
JSONObject imageWatermark = new JSONObject();
imageWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
imageWatermark.put("Type", "Image");
imageWatermark.put("InputFile", imageWatermarkInput);
imageWatermark.put("ReferPos", "TopRight");
imageWatermark.put("Width", "0.05");
imageWatermark.put("Dx", "0");
imageWatermark.put("Dy", "0");
// Text Watermark
JSONObject textConfig = new JSONObject();
textConfig.put("Content", "5rWL6K+V5paH5a2X5rC05Y2w");
textConfig.put("FontName", "SimSun");
textConfig.put("FontSize", "16");
textConfig.put("FontColor", "Red");
textConfig.put("FontAlpha", "0.5");
textConfig.put("Top", "10");
textConfig.put("Left", "10");
JSONObject textWatermark = new JSONObject();
textWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
textWatermark.put("Type", "Text");
textWatermark.put("TextWaterMark", textConfig.toJSONString());
// Video Watermark
JSONObject videoWatermarkInput = new JSONObject();
videoWatermarkInput.put("Location", ossLocation);
videoWatermarkInput.put("Bucket", ossBucket);
try {
    videoWatermarkInput.put("Object", URLEncoder.encode(videoWatermarkObject, "
utf-8"));
} catch (UnsupportedEncodingException e) {

```

```

        throw new RuntimeException("videoWatermark Input URL encode failed");
    }
    JSONObject videoWatermark = new JSONObject();
    videoWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
    videoWatermark.put("Type", "Image");
    videoWatermark.put("InputFile", videoWatermarkInput);
    videoWatermark.put("ReferPos", "BottomLeft");
    videoWatermark.put("Height", "240");
    videoWatermark.put("Dx", "0");
    videoWatermark.put("Dy", "0");
    // Output->Watermarks
    JSONArray watermarks = new JSONArray();
    watermarks.add(imageWatermark);
    watermarks.add(textWatermark);
    watermarks.add(videoWatermark);
    output.put("WaterMarks", watermarks.toJSONString());
    // Outputs
    JSONArray outputs = new JSONArray();
    outputs.add(output);
    request.setOutputBucket(ossBucket);
    request.setOutputLocation(ossLocation);
    // PipelineId
    request.setPipelineId(pipelineId);
    // call api
    SubmitJobsResponse response;
    try {
        response = client.getAcsResponse(request);
        System.out.println("RequestId is:" + response.getRequestId());
        if (response.getJobResultList().get(0).getSuccess()) {
            System.out.println("JobId is:" + response.getJobResultList().get(0).getJob().
getJobId());
        } else {
            System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0)
).getCode() +
                " message:" + response.getJobResultList().get(0).getMessage());
        }
    } catch (ServerException e) {
        e.printStackTrace();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}
}
}

```

3.3.6 Screenshot

1. Create an AcsClient instance.

```

DefaultProfile profile = DefaultProfile.getProfile(
    mpsRegionId, // Region ID
    accessKeyId, // AccessKey ID
    accessKeySecret); // Access Key Secret

```

```
IAcClient client = new DefaultAcClient(profile);
```

2. Create a request and set parameters.

```
SubmitSnapshotJobRequest request = new SubmitSnapshotJobRequest();
```

3. Set the screenshot parameters.

- Input

```
JSONObject input = new JSONObject();
input.put("Location", ossLocation);
input.put("Bucket", ossBucket);
try {
    input.put("Object", URLEncoder.encode(ossInputObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("input URL encode failed");
}
request.setInput(input.toJSONString());
```

- SnapshotConfig

```
JSONObject snapshotConfig = new JSONObject();
```

- OutputFile

```
JSONObject output = new JSONObject();
output.put("Location", ossLocation);
output.put("Bucket", ossBucket);
try {
    output.put("Object", URLEncoder.encode(ossOutputObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("output URL encode failed");
}
snapshotConfig.put("OutputFile", output.toJSONString());
```

- Time

```
snapshotConfig.put("Time", "2");
```

- Interval/Num

```
snapshotConfig.put("Interval", "2");
snapshotConfig.put("Num", "3");
```

- Width/Height

```
snapshotConfig.put("Height", "360");
```

4. Initiate an API request and display the response value.

```
SubmitSnapshotJobResponse response;
response = client.getAcResponse(request);
System.out.println("RequestId is:" + response.getRequestId());
System.out.println("JobId is:" + response.getSnapshotJob().getJobId());
System.out.println(String.format(
    "http://%s.%s.aliyuncs.com/output_00001.jpg",
    ossBucket,
```

```

        ossLocation));
System.out.println(String.format(
    "http://%s.%s.aliyuncs.com/output_00002.jpg",
    ossBucket,
    ossLocation));
System.out.println(String.format(
    "http://%s.%s.aliyuncs.com/output_00003.jpg",
    ossBucket,
    ossLocation));

```

Full code

```

package com.aliyun.mts;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class Snapshot {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "xxx";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output_{Count}.jpg";
    public static void main(String[] args) {
        // DefaultAcsClient
        DefaultProfile profile = DefaultProfile.getProfile(
            mpsRegionId, // Region ID
            accessKeyId, // AccessKey ID
            accessKeySecret); // Access Key Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // request
        SubmitSnapshotJobRequest request = new SubmitSnapshotJobRequest();
        // Input
        JSONObject input = new JSONObject();
        input.put("Location", ossLocation);
        input.put("Bucket", ossBucket);
        try {
            input.put("Object", URLEncoder.encode(ossInputObject, "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("input URL encode failed");
        }
        request.setInput(input.toJSONString());
        // SnapshotConfig
        JSONObject snapshotConfig = new JSONObject();
        // SnapshotConfig->OutputFile
        JSONObject output = new JSONObject();
        output.put("Location", ossLocation);
        output.put("Bucket", ossBucket);
        try {
            output.put("Object", URLEncoder.encode(ossOutputObject, "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("output URL encode failed");
        }
        snapshotConfig.put("OutputFile", output.toJSONString());
    }
}

```

```
// SnapshotConfig->Time
snapshotConfig.put("Time", "2");
// SnapshotConfig->Interval/Num
snapshotConfig.put("Interval", "2");
snapshotConfig.put("Num", "3");
// SnapshotConfig->Width/Height
snapshotConfig.put("Height", "360");
// SnapshotConfig
request.setSnapshotConfig(snapshotConfig.toJSONString());
// PipelineId
request.setPipelineId(pipelineId);
// call api
SubmitSnapshotJobResponse response;
try {
    response = client.getAcsResponse(request);
    System.out.println("RequestId is:" + response.getRequestId());
    System.out.println("JobId is:" + response.getSnapshotJob().getId());
    System.out.println(String.format(
        "http://%s.%s.aliyuncs.com/output_00001.jpg",
        ossBucket,
        ossLocation));
    System.out.println(String.format(
        "http://%s.%s.aliyuncs.com/output_00002.jpg",
        ossBucket,
        ossLocation));
    System.out.println(String.format(
        "http://%s.%s.aliyuncs.com/output_00003.jpg",
        ossBucket,
        ossLocation));
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

3.3.7 HLS data encryption

Scenario

- HLS standard data encryption applies to "protect video protection", it can prevent illegal downloads and illegal dissemination.

Limits

- To use HLS standard data encryption, you must use **SubmitJobs** API.
- HLS standard data encryption function cannot be used in workflow.

Code example

- For more information about MPS SDK, [Installation](#).
- Other code examples

```
<dependency>
<groupId>com.alibaba</groupId>
<artifactId>fastjson</artifactId>
<version>1.2.25</version>
```



```

</dependency>
<dependency>
  <groupId>commons-codec</groupId>
  <artifactId>commons-codec</artifactId>
  <version>1.9</version>
</dependency>

```

Code example

```

package com.aliyun
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.mts.model.v20140618.QueryJobListRequest;
import com.aliyuncs.mts.model.v20140618.QueryJobListResponse;
import com.aliyuncs.mts.model.v20140618.SubmitJobsRequest;
import com.aliyuncs.mts.model.v20140618.SubmitJobsResponse;
import com.aliyuncs.profile.DefaultProfile;
import org.apache.commons.codec.binary.Base64;
public class DataEncryptionDemo {
  private DefaultAcsClient client = null;
  private final String REGION = "cn-shanghai"; //set as needed
  private final String MTS_ENDPOINT = "mts.cn-shanghai.aliyuncs.com"; //set as needed
  private final String ID="idid"; //set as needed
  private final String KEY="keykey"; //set as needed
  private final String LOCATION = "oss-cn-shanghai"; //set as needed
  private final String INPUT_BUCKET = "input-bucket"; //set as needed
  private final String OUTPUT_BUCKET = "output-bucket"; //set as needed
  private final String PIPELINE_ID = "pipelineid"; //set as needed
  public DataEncryptionDemo() throws ClientException {
    DefaultProfile.addEndpoint(REGION, REGION, "Mts", MTS_ENDPOINT);
    this.client = new DefaultAcsClient(DefaultProfile.getProfile(REGION, ID, KEY));
  }
  private JSONObject getInputFile() {
    JSONObject inputFile = new JSONObject();
    inputFile.put("Location", LOCATION);
    inputFile.put("Bucket", INPUT_BUCKET);
    inputFile.put("Object", "uploadvideo/test.flv");
    return inputFile;
  }
  private JSONArray getOutputs() {
    JSONArray outputs = new JSONArray();
    outputs.add(getOutput());
    return outputs;
  }
  private JSONObject getOutput() {
    JSONObject output = new JSONObject();
    output.put("OutputObject", "BaseTest/hls-encryption.m3u8");
    output.put("TemplateId", "S00000001-100020");
    output.put("Encryption", getEncryptionConfigs());
    return output;
  }
  private JSONObject getEncryptionConfigs() {
    JSONObject encryption = new JSONObject();
    encryption.put("Type", "hls-aes-128");
    encryption.put("Key", Base64.encodeBase64URLSafeString("encryptionkey123".
getBytes()));
    encryption.put("KeyUri", Base64.encodeBase64URLSafeString("http://demo.aliyuncs
.com/document/hls128.key".getBytes()));
    encryption.put("KeyType", "Base64");
    return encryption;
  }
}

```

```

private String submitJobs() throws ClientException {
    JSONObject inputFile = getInputFile();
    SubmitJobsRequest request = new SubmitJobsRequest();
    request.setInput(inputFile.toJSONString());
    request.setOutputLocation(LOCATION);
    request.setOutputBucket(OUTPUT_BUCKET);
    request.setOutputs(getOutputs().toJSONString());
    request.setPipelineId(PIPELINE_ID);
    SubmitJobsResponse reponse = this.client.getAcResponse(request);
    System.out.println(JSONObject.toJSONString(reponse.getJobResultList()));
    return reponse.getJobResultList().get(0).getJob().getJobId();
}
public static void main(String[] args) throws ClientException {
    DataEncryptionDemo demo = new DataEncryptionDemo();
    String jobId= demo.submitJobs();
}
}

```

3.3.8 Create HLS standard encryption workflow

Overview

This document is an example of calling API to create HLS standard encryption workflow. For more information about creating workflow, see [AddMediaWorkflow](#).

Code dependency example

- For more information about MPS SDK, see [Installation](#).
- Other dependency.

```

<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>fastjson</artifactId>
  <version>1.2.25</version>
</dependency>

```

Code example

```

package com.aliyun.smallcode;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.mts.model.v20140618.AddMediaWorkflowRequest;
import com.aliyuncs.mts.model.v20140618.AddMediaWorkflowResponse;
import com.aliyuncs.profile.DefaultProfile;
public class MediaHls {
    final String REGION_ID = "<region>";
    final String ACCESS_KEY_ID = "<accessKeyId>";
    final String ACCESS_KEY_SECRET = "<accessKeySecret>";
    final String PIPELINE_ID = "<PipelineId>";
    final String TEMPLATE_ID = "S00000001-100020"; //Transcoding template ID, m3u8
    template, set as needed
    final String OSS_LOCATION = "<OssLocation>";
    final String INPUT_BUCKET = "<InputBucket>"; //Enter bucket
    final String INPUT_PATH = "<InputPath>"; //"HLS-Encryption"
    final String OUTPUT_BUCKET = "<OutputBucket>"; //output bucket
    final String ENCRYPTION_TYPE = "hls-aes-128";
}

```

```

final String HLS_KEY_URI = "<Decryption key URI>"; //http://decrypt.testdomain.com
final String ACT_START = "Act-Start";
final String ACT_ENCRYPTION = "Act-HLS-Encryption";
final String ACT_REPORT = "Act-Report";
private DefaultAcsClient client;
public MediaHls() {
    DefaultProfile profile = DefaultProfile.getProfile(REGION_ID, ACCESS_KEY_ID,
ACCESS_KEY_SECRET);
    this.client = new DefaultAcsClient(profile);
}
public AddMediaWorkflowResponse addMediaWorkflow() throws ClientException {
    AddMediaWorkflowRequest request = new AddMediaWorkflowRequest();
    request.setTopology(createWorkflow().toJSONString());
    request.setName("HLS encryption workflow");
    return this.client.getAcsResponse(request);
}
private JSONObject createWorkflow() {
    JSONObject workflow = new JSONObject();
    JSONObject activities = new JSONObject();
    activities.put(ACT_START, createStartActivity());
    activities.put(ACT_ENCRYPTION, createTrancodeActivity());
    activities.put(ACT_REPORT, createReportActivity());
    workflow.put("Activities", activities);
    workflow.put("Dependencies", createDependencies());
    return workflow;
}
private JSONObject createStartActivity() {
    JSONObject startActivity = new JSONObject();
    startActivity.put("Name", ACT_START);
    startActivity.put("Type", "Start");
    startActivity.put("Parameters", buildStartParameters());
    return startActivity;
}
private JSONObject buildStartParameters() {
    JSONObject parameters = new JSONObject();
    parameters.put("PipelineId", PIPELINE_ID);
    parameters.put("InputFile", buildInputFile());
    return parameters;
}
private JSONObject buildInputFile() {
    JSONObject inputFile = new JSONObject();
    inputFile.put("Bucket", INPUT_BUCKET);
    inputFile.put("Location", OSS_LOCATION);
    inputFile.put("ObjectPrefix", INPUT_PATH);
    return inputFile;
}
private JSONObject createTrancodeActivity() {
    JSONObject transcodeActivity = new JSONObject();
    transcodeActivity.put("Name", ACT_ENCRYPTION);
    transcodeActivity.put("Type", "Transcode");
    transcodeActivity.put("Parameters", buildTranscodeParameters());
    return transcodeActivity;
}
private JSONObject buildTranscodeParameters() {
    JSONObject transcodeParamters = new JSONObject();
    transcodeParamters.put("OutputBucket", OUTPUT_BUCKET);
    transcodeParamters.put("OutputLocation", OSS_LOCATION);
    transcodeParamters.put("Outputs", buildOutputsConfig());
    return transcodeParamters;
}
private JSONArray buildOutputsConfig() {
    JSONArray outputs = new JSONArray();
    JSONObject output = new JSONObject();
    output.put("ObjectRegex", ACT_ENCRYPTION +("/{RunId}/{FileName}");
}

```

```

        output.put("TemplateId", TEMPLATE_ID);
        output.put("Encryption", buildEncryption());
        outputs.add(output);
        return outputs;
    }
    private JSONObject buildEncryption() {
        JSONObject encryption = new JSONObject();
        encryption.put("Type", ENCRYPTION_TYPE);
        encryption.put("KeyUri", HLS_KEY_URI);
        return encryption;
    }
    private JSONObject createReportActivity() {
        JSONObject reportActivity = new JSONObject();
        reportActivity.put("Name", ACT_REPORT);
        reportActivity.put("Parameters", new JSONObject());
        reportActivity.put("Type", "Report");
        return reportActivity;
    }
    private JSONObject createDependencies() {
        JSONObject dependencies = new JSONObject();
        JSONArray subActivityOfStart = new JSONArray();
        subActivityOfStart.add(ACT_ENCRYPTION);
        dependencies.put(ACT_START, subActivityOfStart);
        JSONArray subActivityOfTranscode = new JSONArray();
        subActivityOfTranscode.add(ACT_REPORT);
        dependencies.put(ACT_ENCRYPTION, subActivityOfTranscode);
        dependencies.put(ACT_REPORT, new JSONArray());
        return dependencies;
    }
    public static void main(String[] args) throws ClientException {
        MediaHls mediaHls = new MediaHls();
        AddMediaWorkflowResponse response = mediaHls.addMediaWorkflow();
        System.out.println(JSONObject.toJSONString(response));
    }
}

```

3.3.9 MPS queues management

The system automatically creates an MPS queue when you open the MPS service. You can also manage the MPS queue (pipeline) by using interfaces. For example: **SearchPipeline**, **QueryPipelineList**, **UpdatePipeline**.

Search MPS queue

You can search the MPS queue information by using the **SearchPipeline** interface.

```

String region = "<region>";
String accessKeyId = "<accessKeyId>";
String accessKeySecret = "<accessKeySecret>";
DefaultProfile profile = DefaultProfile.getProfile(region, accessKeyId, accessKeySecret);
DefaultAcsClient client = new DefaultAcsClient(profile);
SearchPipelineRequest request = new SearchPipelineRequest();
// If an error occurs, it can throw ClientException or ServerException.
SearchPipelineResponse response = client.getAcsResponse(request);
List<SearchPipelineResponse.Pipeline> pipelines = response.getPipelineList();
for (SearchPipelineResponse.Pipeline pipeline : pipelines) {
    System.out.println("pipeline id:" + pipeline.getId() + ", name:" + pipeline.getName()
+ ", state:" + pipeline.getState());
}

```

```
}

```

Query MPS queue

If you know the pipelineId, you can use pipelineId to call **QueryPipelineList** interface to query MPS queue information.

```
String region = "<region>";
String accessKeyId = "<accessKeyId>";
String accessKeySecret = "<accessKeySecret>";
// known pipeline ID, seperated by comma (,)
String pipelineIds = "<pipelineIds>";
DefaultProfile profile = DefaultProfile.getProfile(region, accessKeyId, accessKeySecret);
DefaultAcsClient client = new DefaultAcsClient(profile);
QueryPipelineListRequest request = new QueryPipelineListRequest();
request.setPipelineIds(pipelineId);
//If an error occurs, it can throw ClientException or ServerException
QueryPipelineListResponse response = client.getAcsResponse(request);
List<QueryPipelineListResponse.Pipeline> pipelines = response.getPipelineList();
for (QueryPipelineListResponse.Pipeline pipeline : pipelines) {
    System.out.println("pipeline id:" + pipeline.getId() + ", name:" + pipeline.getName()
+ ", state:" + pipeline.getState());
}
```

Update MPS queue

Use the **UpdatePipeline** interface to update MPS queue information, including MPS queue name and status. The status includes Active and Paused.

```
String region = "<region>";
String accessKeyId = "<accessKeyId>";
String accessKeySecret = "<accessKeySecret>";
DefaultProfile profile = DefaultProfile.getProfile(region, accessKeyId, accessKeySecret);
DefaultAcsClient client = new DefaultAcsClient(profile);
// Search user pipeline, the number is one by default.
SearchPipelineRequest searchPipelineRequest = new SearchPipelineRequest();
// If an error occurs, it can throw ClientException or ServerException.
SearchPipelineResponse searchPipelineResponse = client.getAcsResponse(searchPipelineRequest);
List<SearchPipelineResponse.Pipeline> pipelines = searchPipelineResponse.getPipelineList();
SearchPipelineResponse.Pipeline queryPipeline = pipelines.get(0);
//Update MPS queue status
UpdatePipelineRequest request = new UpdatePipelineRequest();
request.setPipelineId(queryPipeline.getId());
request.setState("Paused");
request.setName(queryPipeline.getName());
//If an error occurs, it can throw ClientException.
UpdatePipelineResponse response = client.getAcsResponse(request);
UpdatePipelineResponse.Pipeline pipeline = response.getPipeline();
```

```
System.out.println("pipeline id:" + pipeline.getId() + ", name:" + pipeline.getName() +
", state:" + pipeline.getState());
```

3.3.10 Query media - Use OSS file address

If you do not know the media ID (a live video converted to an on-demand video using the media workflow), you can use the media input URL to query the media information over

QueryMediaListByURL.

```
package com.aliyun.mts.api.demo
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.mts.model.v20140618.QueryMediaListByURLRequest;
import com.aliyuncs.mts.model.v20140618.QueryMediaListByURLResponse;
import com.aliyuncs.profile.DefaultProfile;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
public class QueryMediaListByURLDemo {
    private DefaultAcsClient client = null;
    private final String REGION = "<region>";
    private final String ID="<accessKeyId>";
    private final String KEY ="<accessKeySecret>";
    public QueryMediaListByURLDemo() throws ClientException {
        this.client = new DefaultAcsClient(DefaultProfile.getProfile(REGION, ID, KEY));
    }
    //Query the media information, such as the media ID, media status, and other
    attributes, based on the OSS URL of the media source
    private void queryMediaListByURL() throws ClientException, UnsupportedEncodingE
xception {
        QueryMediaListByURLRequest request = new QueryMediaListByURLRequest();
        String ossHost = 'http://<input-bucket>.<region>.aliyuncs.com/';
        String ossObject = "test/The Legend of the Swordsman.mp4";
        //ossObject must be RFC3986-encoded
        String rfc3986Object = encodeByRFC3986(ossObject);
        request.setFileURLs(ossHost + rfc3986Object);
        QueryMediaListByURLResponse response = this.client.getAcsResponse(request);
        System.out.println(JSONObject.toJSONString(response.getMediaList()));
    }
    private String encodeByRFC3986(String object) throws UnsupportedEncodingException
    {
        StringBuilder builder = new StringBuilder();
        String[] segments = object.split("/");
        for (int i = 0; i < segments.length; i++) {
            builder.append(percentEncode(segments[i]));
            if (i != segments.length - 1) {
                builder.append("/");
            }
        }
        return builder.toString();
    }
    private static String percentEncode(String value) throws UnsupportedEncodingE
xception {
        if (value == null)
            return null;
        return URLEncoder.encode(value, "UTF-8").replace("+", "%20").replace("*", "%2A").
replace("%7E", "~");
    }
    public static void main(String[] args) {
        try {
            QueryMediaListByURLDemo demo = new QueryMediaListByURLDemo();
```

```

        demo.queryMediaListByURL();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

3.3.11 Add media

Add media file to Media Files, and the user can specify workflow ID to trigger the workflow, which then processes the video file:

```

package com.aliyun.mts;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.AddMediaRequest;
import com.aliyuncs.mts.model.v20140618.AddMediaResponse;
import com.aliyuncs.profile.DefaultProfile;
import org.apache.commons.lang.exception.ExceptionUtils;
public class AddMedia {
    //Step 1 .set region: cn-hangzhou、cn-shenzhen、cn-shanghai、cn-beijing
    private static final String REGION = "cn-shenzhen";
    private static final String OSS_REGION = "oss-cn-shenzhen";
    private static final String mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
    //Step 2.set accesskey & keySecret
    private static String accessKeyId = "";
    private static String accessKeySecret = "";
    private static DefaultAcsClient aliyunClient;
    static {
        try {
            DefaultProfile.addEndpoint(REGION, REGION, "Mts", mtsEndpoint);
        } catch (ClientException e) {
            System.out.print(ExceptionUtils.getStackTrace(e));
            System.exit(1);
        }
        aliyunClient = new DefaultAcsClient(DefaultProfile.getProfile(REGION, accessKeyId
, accessKeySecret));
    }
    public static void main(String[] args) throws ClientException {
        AddMediaRequest request = new AddMediaRequest();
        request.setFileURL("http://mtb-sz-in.oss-cn-shenzhen.aliyuncs.com/media/r180-
ABC.mp4");
        request.setMediaWorkflowId("829bed0300994057a49e4f16de957e34");
        try {
            AddMediaResponse response = aliyunClient.getAcsResponse(request);
            System.out.println(JSONObject.toJSONString(response));
        } catch (ServerException e) {
            System.out.println("Code:" + e.getErrCode() + " Msg:" + e.getMessage());
        } catch (ClientException e) {
            System.out.println("Code:" + e.getErrCode() + " Msg:" + e.getMessage());
        }
    }
}

```

```
}

```

3.3.12 Splicing and simple cutting

1. Create AcsClient instance.

```
DefaultProfile profile = DefaultProfile.getProfile(
    mpsRegionId, // Region ID
    accessKeyId, // AccessKey ID
    accessKeySecret); // Access Key Secret
IAcsClient client = new DefaultAcsClient(profile);

```

2. Create request and set parameters.

```
SubmitJobsRequest request = new SubmitJobsRequest();

```

3. Set transcoding parameters.

- Input

```
JSONObject input = new JSONObject();
input.put("Location", ossLocation);
input.put("Bucket", ossBucket);
try {
    input.put("Object", URLEncoder.encode(headObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("input URL encode failed");
}
request.setInput(input.toJSONString());

```

- Output

```
String outputOSSObject;
try {
    outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8");
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("output URL encode failed");
}
JSONObject output = new JSONObject();
output.put("OutputObject", outputOSSObject);
// Ouput->TemplateId
output.put("TemplateId", templateId);

```

- Video

```
JSONObject video = new JSONObject();
video.put("Width", "1280");
video.put("Height", "720");
output.put("Video", video.toJSONString());

```

- MergeList

```
JSONObject mergeVideo = new JSONObject();
String mergeVideoURL;
try {
    mergeVideoURL = String.format(
        "http://%.s.%.s.aliyuncs.com/%.s",
        ossBucket,

```



```

        ossLocation,
        URLEncoder.encode(ossInputObject, "utf-8"));
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("mergeVideoURL encode failed");
    }
    mergeVideo.put("MergeURL", mergeVideoURL);
    JSONObject mergeTail = new JSONObject();
    String mergeTailURL;
    try {
        mergeTailURL = String.format(
            "http://%s.%s.aliyuncs.com/%s",
            ossBucket,
            ossLocation,
            URLEncoder.encode(tailObject, "utf-8"));
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("mergeTailURL encode failed");
    }
    mergeTail.put("MergeURL", mergeTailURL);
    JSONArray mergeList = new JSONArray();
    mergeList.add(mergeVideo);
    mergeList.add(mergeTail);
    output.put("MergeList", mergeList.toJSONString());

```

4. Initiate API request and display returned value.

```

SubmitJobsResponse response;
response = client.getAcsResponse(request);
System.out.println("RequestId is:" + response.getRequestId());
if (response.getJobResultList().get(0).getSuccess()) {
    System.out.println("JobId is:" + response.getJobResultList().get(0).getJob().getJobId());
} else {
    System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0).
        getCode() +
        " message:" + response.getJobResultList().get(0).getMessage());
}

```

Full codes

```

package com.aliyun.mts;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class Merge {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String templateId = "S00000001-200030";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "xxx";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output.mp4";
    private static String headObject = "head.mp4";
    private static String tailObject = "tail.mp4";

```

```
public static void main(String[] args) {
    // DefaultAcsClient
    DefaultProfile profile = DefaultProfile.getProfile(
        mpsRegionId, // Region ID
        accessKeyId, // AccessKey ID
        accessKeySecret); // Access Key Secret
    IAcsClient client = new DefaultAcsClient(profile);
    // request
    SubmitJobsRequest request = new SubmitJobsRequest();
    // Input
    JSONObject input = new JSONObject();
    input.put("Location", ossLocation);
    input.put("Bucket", ossBucket);
    try {
        input.put("Object", URLEncoder.encode(headObject, "utf-8"));
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("input URL encode failed");
    }
    request.setInput(input.toJSONString());
    // Output
    String outputOSSObject;
    try {
        outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8");
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("output URL encode failed");
    }
    JSONObject output = new JSONObject();
    output.put("OutputObject", outputOSSObject);
    // Ouput->TemplateId
    output.put("TemplateId", templateId);
    // Ouput->Video
    JSONObject video = new JSONObject();
    video.put("Width", "1280");
    video.put("Height", "720");
    output.put("Video", video.toJSONString());
    // Output->MergeList
    JSONObject mergeVideo = new JSONObject();
    String mergeVideoURL;
    try {
        mergeVideoURL = String.format(
            "http://%s.%s.aliyuncs.com/%s",
            ossBucket,
            ossLocation,
            URLEncoder.encode(ossInputObject, "utf-8"));
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("mergeVideoURL encode failed");
    }
    mergeVideo.put("MergeURL", mergeVideoURL);
    JSONObject mergeTail = new JSONObject();
    String mergeTailURL;
    try {
        mergeTailURL = String.format(
            "http://%s.%s.aliyuncs.com/%s",
            ossBucket,
            ossLocation,
            URLEncoder.encode(tailObject, "utf-8"));
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("mergeTailURL encode failed");
    }
    mergeTail.put("MergeURL", mergeTailURL);
    JSONArray mergeList = new JSONArray();
    mergeList.add(mergeVideo);
    mergeList.add(mergeTail);
    output.put("MergeList", mergeList.toJSONString());
}
```

```

// Outputs
JSONArray outputs = new JSONArray();
outputs.add(output);
request.setOutputs(outputs.toJSONString());
request.setOutputBucket(ossBucket);
request.setOutputLocation(ossLocation);
// PipelineId
request.setPipelineId(pipelineId);
// call api
SubmitJobsResponse response;
try {
    response = client.getAcsResponse(request);
    System.out.println("RequestId is:" + response.getRequestId());
    if (response.getJobResultList().get(0).getSuccess()) {
        System.out.println("JobId is:" + response.getJobResultList().get(0).getJob().
getJobId());
    } else {
        System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0)
).getCode() +
        " message:" + response.getJobResultList().get(0).getMessage());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
}

```

3.3.13 Add media workflow

The user can assemble activities provided by MPS, such as transcode activity and screenshot activity into a topology. The topology is as follows:

Topology type

```

package com.aliyun.mts;
import com.alibaba.fastjson.annotation.JSONField;
import java.util.List;
import java.util.Map;
public class Topology {
    @JSONField(name = "Activities")
    private Map<String, Activity> activities;
    @JSONField(name = "Dependencies")
    private Map<String, List<String>> dependencies;
    public Map<String, List<String>> dependencies() {
        return this.getDependencies();
    }
    public Topology() {
    }
    public Topology(Map<String, Activity> activities, Map<String, List<String>> dependencies) {
        this.setActivities(activities);
        this.setDependencies(dependencies);
    }
    public Map<String, Activity> getActivities() {
        return activities;
    }
    public Map<String, List<String>> getDependencies() {
        return dependencies;
    }
}

```

```

public void setActivities(Map<String, Activity> activities) {
    this.activities = activities;
}
public void setDependencies(Map<String, List<String>> dependencies) {
    this.dependencies = dependencies;
}
}

```

ActivityType enumeration type

```

package com.aliyun.mts;
/**
 * Created by zhongyizengzy on 18/3/22.
 */
public enum ActivityType {
    Start, Transcode, Snapshot, MediaInfo, Analysis, Cover, Summary, Censor, Report,
    UploadVerify, GenerateMasterPlayList, AudioGroup, SubtitleGroup, PackageConfig
}

```

Activity type

```

package com.aliyun.mts;
import com.alibaba.fastjson.annotation.JSONField;
import java.util.Map;
public class Activity {
    @JSONField(name = "Type")
    private String type;
    @JSONField(name = "Parameters")
    private Map<String, String> parameters;
    public Activity() {
    }
    public Map<String, String> parameters() {
        return this.getParameters();
    }
    public Activity(String type, Map<String, String> parameters) {
        this.setType(type);
        this.setParameters(parameters);
    }
    public String getType() {
        return type;
    }
    public Map<String, String> getParameters() {
        return parameters;
    }
    public void setType(String type) {
        this.type = type;
    }
    public void setParameters(Map<String, String> parameters) {
        this.parameters = parameters;
    }
}

```

AddMediaWorkflow type

```

package com.aliyun.mts;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.AddMediaWorkflowRequest;

```

```

import com.aliyuncs.mts.model.v20140618. AddMediaWorkflowResponse;
import com.aliyuncs.profile.DefaultProfile;
import org.apache.commons.lang.exception.ExceptionUtils;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
public class AddMediaWorkflow {
    //Step 1 .set region: cn-hangzhou、cn-shenzhen、cn-shanghai、cn-beijing
    private static final String REGION = "cn-shenzhen";
    private static final String OSS_REGION = "oss-cn-shenzhen";
    private static final String mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
    //Step 2.set accesskey & keySecret
    private static String accessKeyId = "";
    private static String accessKeySecret = "";
    //Step 3.set mps transcoding queue id
    private static String PIPELINE_ID = "38bba54d524448be92d277caaa8da118";
    private static DefaultAcsClient aliyunClient;
    static {
        try {
            DefaultProfile.addEndpoint(REGION, REGION, "Mts", mtsEndpoint);
        } catch (ClientException e) {
            System.out.print(ExceptionUtils.getStackTrace(e));
            System.exit(1);
        }
        aliyunClient = new DefaultAcsClient(DefaultProfile.getProfile(REGION, accessKeyId,
accessKeySecret));
    }
    public static void main(String[] args) throws ClientException {
        AddMediaWorkflowRequest request = new AddMediaWorkflowRequest();
        request.setName("Sequential-workflow");
        Topology topology = new Topology();
        HashMap<String, Activity> activities = new HashMap<String, Activity>();
        Activity startNode = new Activity();
        startNode.setType(ActivityType.Start.name());
        HashMap<String, String> startNodeParameters = new HashMap<String, String>();
        JSONObject inputFile = new JSONObject();
        inputFile.put("Bucket", "mtb-sz-in");
        inputFile.put("Location", OSS_REGION);
        inputFile.put("ObjectPrefix", "media/");
        startNodeParameters.put("InputFile", inputFile.toString());
        startNodeParameters.put("PipelineId", PIPELINE_ID);
        startNode.setParameters(startNodeParameters);
        activities.put("startNode", startNode);
        Activity transcode = new Activity();
        transcode.setType(ActivityType.Transcode.name());
        HashMap<String, String> transcodingParameters = new HashMap<String, String>();
        JSONArray outputs = new JSONArray();
        JSONObject output = new JSONObject();
        try {
            output.put("OutputObject", URLEncoder.encode("transcode/{ObjectPrefix}/{
FileName}.{ ExtName}", "UTF-8"));
        } catch (UnsupportedEncodingException e) {
            System.exit(1);
        }
        output.put("TemplateId", "S00000001-000070");
        outputs.add(output);
        transcodingParameters.put("Outputs", outputs.toJSONString());
        transcodingParameters.put("OutputBucket", "mtb-sz-out");
        transcodingParameters.put("OutputLocation", OSS_REGION);
        transcode.setParameters(transcodingParameters);
        activities.put("transcodingNode", transcode);
    }
}

```

```

Activity report = new Activity();
report.setType(ActivityType.Report.name());
HashMap<String, String> reportParameters = new HashMap<String, String>();
report.setParameters(reportParameters);
activities.put("reportNode", report);
topology.setActivities(activities);
HashMap<String, List<String>> dependencies = new HashMap<String, List<String>>();
dependencies.put("startNode", Arrays.asList("transcodingNode"));
dependencies.put("transcodingNode", Arrays.asList("reportNode"));
dependencies.put("reportNode", new ArrayList<String>());
topology.setDependencies(dependencies);
request.setTopology(JSONObject.toJSONString(topology));
try {
    AddMediaWorkflowResponse response = aliyunClient.getAcResponse(request);
    System.out.println(JSONObject.toJSONString(response));
} catch (ServerException e) {
    System.out.println("Code:" + e.getErrCode() + " Msg:" + e.getMessage());
} catch (ClientException e) {
    System.out.println("Code:" + e.getErrCode() + " Msg:" + e.getMessage());
}
}
}

```

3.3.14 Splicing-opening and ending scenes

1. Create AcClient instance.

```

DefaultProfile profile = DefaultProfile.getProfile(
    mpsRegionId, // Region ID
    accessKeyId, // AccessKey ID
    accessKeySecret); // Access Key Secret
IAcsClient client = new DefaultAcsClient(profile);

```

2. Create request and set parameters.

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. Set transcoding parameters.

- Input

```

JSONObject input = new JSONObject();
input.put("Location", ossLocation);
input.put("Bucket", ossBucket);
try {
    input.put("Object", URLEncoder.encode(headObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("input URL encode failed");
}
request.setInput(input.toJSONString());

```

- Output

```

String outputOSSObject;
try {
    outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8");
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("output URL encode failed");
}
JSONObject output = new JSONObject();

```

```
output.put("OutputObject", outputOSSObject);
// Ouput->TemplateId
output.put("TemplateId", templateId);
```

- Video

```
JSONObject video = new JSONObject();
video.put("Width", "1280");
video.put("Height", "720");
output.put("Video", video.toJSONString());
```

- OpeningList

```
JSONObject openingVideo = new JSONObject();
String openingVideoURL;
try {
    openingVideoURL = String.format(
        "http://%.%.aliyuncs.com/%",
        ossBucket,
        ossLocation,
        URLEncoder.encode(headObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("mergeVideoURL encode failed");
}
openingVideo.put("OpenUrl", openingVideoURL);
openingVideo.put("Width", "640");
openingVideo.put("Start", "2");
JSONArray openingVideoList = new JSONArray();
openingVideoList.add(openingVideo);
output.put("OpeningList", openingVideoList.toJSONString());
```

- TailSlateList

```
JSONObject tailSlateVideo = new JSONObject();
String tailSlateVideoURL;
try {
    tailSlateVideoURL = String.format(
        "http://%.%.aliyuncs.com/%",
        ossBucket,
        ossLocation,
        URLEncoder.encode(tailObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("mergeTailURL encode failed");
}
tailSlateVideo.put("TailUrl", tailSlateVideoURL);
tailSlateVideo.put("Width", "640");
tailSlateVideo.put("BlendDuration", "3");
tailSlateVideo.put("BgColor", "Black");
JSONArray tailSlateVideoList = new JSONArray();
tailSlateVideoList.add(tailSlateVideo);
output.put("TailSlateList", tailSlateVideoList.toJSONString());
```

4. Initiate API request and display returned value.

```
SubmitJobsResponse response;
response = client.getAcsResponse(request);
System.out.println("RequestId is:" + response.getRequestId());
if (response.getJobResultList().get(0).getSuccess()) {
    System.out.println("JobId is:" + response.getJobResultList().get(0).getJob().getJobId());
} else {
```

```

        System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0).
        getCode() +
            " message:" + response.getJobResultList().get(0).getMessage());
    }

```

Full codes

```

package com.aliyun.mts;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.model.v20140618.*;
public class OpenTail {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String templateId = "S00000001-200030";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "xxx";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output.mp4";
    private static String headObject = "head.mp4";
    private static String tailObject = "tail.mp4";
    public static void main(String[] args) {
        // DefaultAcsClient
        DefaultProfile profile = DefaultProfile.getProfile(
            mpsRegionId, // Region ID
            accessKeyId, // AccessKey ID
            accessKeySecret); // Access Key Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // request
        SubmitJobsRequest request = new SubmitJobsRequest();
        // Input
        JSONObject input = new JSONObject();
        input.put("Location", ossLocation);
        input.put("Bucket", ossBucket);
        try {
            input.put("Object", URLEncoder.encode(ossInputObject, "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("input URL encode failed");
        }
        request.setInput(input.toJSONString());
        // Output
        String outputOSSObject;
        try {
            outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8");
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("output URL encode failed");
        }
        JSONObject output = new JSONObject();
        output.put("OutputObject", outputOSSObject);
        // Output->TemplateId
        output.put("TemplateId", templateId);
        // Output->OpeningList
        JSONObject openingVideo = new JSONObject();
        String openingVideoURL;

```



```

try {
    openingVideoURL = String.format(
        "http://%s.%s.aliyuncs.com/%s",
        ossBucket,
        ossLocation,
        URLEncoder.encode(headObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("mergeVideoURL encode failed");
}
openingVideo.put("OpenUrl", openingVideoURL);
openingVideo.put("Width", "640");
openingVideo.put("Start", "2");
JSONArray openingVideoList = new JSONArray();
openingVideoList.add(openingVideo);
output.put("OpeningList", openingVideoList.toJSONString());
// Output->TailSlateList
JSONObject tailSlateVideo = new JSONObject();
String tailSlateVideoURL;
try {
    tailSlateVideoURL = String.format(
        "http://%s.%s.aliyuncs.com/%s",
        ossBucket,
        ossLocation,
        URLEncoder.encode(tailObject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("mergeTailURL encode failed");
}
tailSlateVideo.put("TailUrl", tailSlateVideoURL);
tailSlateVideo.put("Width", "640");
tailSlateVideo.put("BlendDuration", "3");
tailSlateVideo.put("BgColor", "Black");
JSONArray tailSlateVideoList = new JSONArray();
tailSlateVideoList.add(tailSlateVideo);
output.put("TailSlateList", tailSlateVideoList.toJSONString());
// Outputs
JSONArray outputs = new JSONArray();
outputs.add(output);
request.setOutputs(outputs.toJSONString());
request.setOutputBucket(ossBucket);
request.setOutputLocation(ossLocation);
// PipelineId
request.setPipelineId(pipelineId);
// call api
SubmitJobsResponse response;
try {
    response = client.getAcsResponse(request);
    System.out.println("RequestId is:" + response.getRequestId());
    if (response.getJobResultList().get(0).getSuccess()) {
        System.out.println("JobId is:" + response.getJobResultList().get(0).getJob().
getJobId());
    } else {
        System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0)
).getCode() +
            " message:" + response.getJobResultList().get(0).getMessage());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}

```

```
}
```

3.4 Python SDK

3.4.1 Overview

MPS Python SDK is based on Alibaba Cloud Python SDK, you must learn about the basic knowledge of Alibaba Cloud Python SDK.

- [Alibaba Cloud SDK Developer Guide > Python SDK > Quick start](#)
- [Alibaba Cloud SDK Developer Guide > Python SDK > User Guide > Use Python SDK](#)
- [Alibaba Cloud Python SDK GitHub](#)

After learning about the basic knowledge, you can perform MPS Python SDK installation. For more information, see [MPS > SDK Reference > Transcoding SDKs > Python SDK > Installation](#).

3.4.2 Installation

This article introduces pip installation method recommended by Alibaba Cloud Python SDK.

- Python 2.x

```
...  
pip install aliyun-python-sdk-core  
pip install aliyun-python-sdk-mts  
...
```

- Python 3.x

```
...  
pip install aliyun-python-sdk-core-v3  
pip install aliyun-python-sdk-mts  
...
```

3.4.3 Quick start

1. Create AcsClient instance.

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. Create request and set parameters.

```
request = SubmitJobsRequest.SubmitJobsRequest()  
request.set_accept_format('json')
```

3. Initiate API request and display returned value.

```
response_str = client.do_action_with_exception(request)  
response = json.loads(response_str)
```

```
print 'PipelineName is:', response['PipelineList']['Pipeline'][0]['Name']
print 'PipelineId is:', response['PipelineList']['Pipeline'][0]['Id']
```

Full code

```
# -*- coding: utf8 -*-
import json
from aliyunsdkcore.client import AcsClient
from aliyunsdkmpts.request.v20140618 import SearchPipelineRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_ids = ['cn-hangzhou', 'cn-beijing', 'cn-shenzhen', 'cn-shanghai',
                  'cn-hongkong', 'us-west-1', 'ap-southeast-1',
                  'ap-northeast-1', 'eu-central-1', 'ap-south-1']
for mps_region_id in mps_region_ids:
    print 'region is:', mps_region_id
    # Create AcsClient instance
    client = AcsClient(access_key_id, access_key_secret, mps_region_id)
    # Create request, and set parameters
    request = SearchPipelineRequest.SearchPipelineRequest()
    request.set_accept_format('json')
    # Initiate API request and display returned value
    response_str = client.do_action_with_exception(request)
    response = json.loads(response_str)
    print 'PipelineName is:', response['PipelineList']['Pipeline'][0]['Name']
    print 'PipelineId is:', response['PipelineList']['Pipeline'][0]['Id']
```

3.4.4 Transcoding

1. Create AcsClient instance.

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. Create request, and set parameters.

```
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
```

3. Transcoding parameters.

- Input

```
job_input = {'Location': oss_location,
             'Bucket': oss_bucket,
             'Object': quote(oss_input_object) }
```

```
request.set_Input(json.dumps(job_input))
```

- Output

```
output = {'OutputObject': quote(oss_output_object)}
```

- Container

```
output['Container'] = {'Format': 'mp4'}
```

- Video

```
output['Video'] = {'Codec': 'H. 264',
                  'Bitrate': 1500,
                  'Width': 1280,
                  'Fps': 25}
```

- Audio

```
output['Audio'] = {'Codec': 'AAC',
                  'Bitrate': 128,
                  'Channels': 2,
                  'Samplerate': 44100}
```

- TemplateId

```
output['TemplateId'] = template_id
```

- PipelineId

```
request.set_PipelineId(pipeline_id)
```

4. Initiate API request and display returned value.

```
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']['JobId']
else:
    print ('SubmitJobs Failed code:',
          response['JobResultList']['JobResult'][0]['Code'],
          ' message:',
          response['JobResultList']['JobResult'][0]['Message'])
```

Full code

```
# -*- coding: utf8 -*-
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitJobsRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
template_id = 'S00000001-200010'
oss_location = 'oss-cn-hangzhou'
```

```

oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output.mp4'
Create AcsClient instance
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
Create request, and set parameters
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
            'Bucket': oss_bucket,
            'Object': quote(oss_input_object) }
request.set_Input(json.dumps(job_input))
# Output
output = {'OutputObject': quote(oss_output_object)}
# Ouput->Container
output['Container'] = {'Format': 'mp4'}
# Ouput->Video
output['Video'] = {'Codec': 'H. 264',
                  'Bitrate': 1500,
                  'Width': 1280,
                  'Fps': 25}
# Ouput->Audio
output['Audio'] = {'Codec': 'AAC',
                  'Bitrate': 128,
                  'Channels': 2,
                  'Samplerate': 44100}
# Ouput->TemplateId
output['TemplateId'] = template_id
outputs = [output]
request.set_Outputs(json.dumps(outputs))
request.set_OutputBucket(oss_bucket)
request.set_OutputLocation(oss_location)
# PipelineId
request.set_PipelineId(pipeline_id)
Initiate API request and diaplay returned value
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']['JobId']
else:
    print ('SubmitJobs Failed code:',
          response['JobResultList']['JobResult'][0]['Code'],
          'message:',
          response['JobResultList']['JobResult'][0]['Message'])

```

3.4.5 Watermarks

1. Create AcsClient instance.

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. Create request, and set parameters.

```
request = SubmitJobsRequest.SubmitJobsRequest()
```

```
request.set_accept_format('json')
```

3. Watermark parameters.

- Image watermark

```
image_watermark_input = {
    'Location': oss_location,
    'Bucket': oss_bucket,
    'Object': quote(image_watermark_object)
}
image_watermark = {
    'WaterMarkTemplateId': watermark_template_id,
    'Type': 'Image',
    'InputFile': image_watermark_input,
    'ReferPos': 'TopRight',
    'Width': 0.05,
    'Dx': 0,
    'Dy': 0
}
```

- Text watermark

```
text_config = {
    'Content': '5rWL6K+V5paH5a2X5rC05Y2w',
    'FontName': 'SimSun',
    'FontSize': 16,
    'FontColor': 'Red',
    'FontAlpha': 0.5,
    'Top': 10,
    'Left': 10
}
text_watermark = {
    'WaterMarkTemplateId': watermark_template_id,
    'Type': 'Text',
    'TextWaterMark': text_config
}
```

- Video watermark

```
video_watermark_input = {
    'Location': oss_location,
    'Bucket': oss_bucket,
    'Object': quote(video_watermark_object)
}
video_watermark = {
    'WaterMarkTemplateId': watermark_template_id,
    'Type': 'Image',
    'InputFile': video_watermark_input,
    'ReferPos': 'BottomLeft',
    'Height': 240,
    'Dx': 0,
    'Dy': 0
}
```

4. Initiate API request and display returned value.

```
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
```

```

print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']['JobId']
else:
    print ('SubmitJobs Failed code:',
          response['JobResultList']['JobResult'][0]['Code'],
          ' message:',
          response['JobResultList']['JobResult'][0]['Message'])

```

Full codes

```

# -*- coding: utf8 -*-
from pprint import pprint
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitJobsRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
watermark_template_id = 'xxx'
template_id = 'S00000001-200030'
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output.mp4'
image_watermark_object = 'logo.png'
video_watermark_object = 'logo.mov'
# AcsClient
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# request
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
             'Bucket': oss_bucket,
             'Object': quote(oss_input_object) }
request.set_input(json.dumps(job_input))
# Output
output = {'OutputObject': quote(oss_output_object)}
# Output->TemplateId
output['TemplateId'] = template_id
## Image Watermark
image_watermark_input = {'Location': oss_location,
                          'Bucket': oss_bucket,
                          'Object': quote(image_watermark_object) }
image_watermark = {
    'WaterMarkTemplateId': watermark_template_id,
    'Type': 'Image',
    'InputFile': image_watermark_input,
    'ReferPos': 'TopRight',
    'Width': 0.05,
    'Dx': 0,
    'Dy': 0
}
## Text Watermark
text_config = {
    'Content': '5rWL6K+V5paH5a2X5rC05Y2w',
    'FontName': 'SimSun',
    'FontSize': 16,
    'FontColor': 'Red',
    'FontAlpha': 0.5,
    'Top': 10,
    'Left': 10
}

```

```

    }
    text_watermark = {
        'WaterMarkTemplateId': watermark_template_id,
        'Type': 'Text',
        'TextWaterMark': text_config
    }
    ## Video Watermark
    video_watermark_input = {'Location': oss_location,
        'Bucket': oss_bucket,
        'Object': quote(video_watermark_object) }
    video_watermark = {
        'WaterMarkTemplateId': watermark_template_id,
        'Type': 'Image',
        'InputFile': video_watermark_input,
        'ReferPos': 'BottomLeft',
        'Height': 240,
        'Dx': 0,
        'Dy': 0
    }
    # Output->Watermarks
    watermarks = [image_watermark, text_watermark, video_watermark]
    output['WaterMarks'] = watermarks
    # Outputs
    outputs = [output]
    request.set_Outputs(json.dumps(outputs))
    request.set_OutputBucket(oss_bucket)
    request.set_OutputLocation(oss_location)
    # PipelineId
    request.set_PipelineId(pipeline_id)
    # call api
    response_str = client.do_action_with_exception(request)
    response = json.loads(response_str)
    print 'RequestId is:', response['RequestId']
    if response['JobResultList']['JobResult'][0]['Success']:
        print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']['JobId']
    else:
        print ('SubmitJobs Failed code:',
            response['JobResultList']['JobResult'][0]['Code'],
            ' message:',
            response['JobResultList']['JobResult'][0]['Message'])

```

3.4.6 Screenshot

1. Create an AcsClient instance.

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. Create request and set parameters.

```
request = SubmitSnapshotJobRequest.SubmitSnapshotJobRequest()
request.set_accept_format('json')
```

3. Set screenshot parameters.

- Input

```
job_input = {'Location': oss_location,
    'Bucket': oss_bucket,
    'Object': quote(head_object) }
```



```
request.set_Input(json.dumps(job_input))
```

- SnapshotConfig

- OutputFile

```
job_output = {'Location': oss_location,
              'Bucket': oss_bucket,
              'Object': quote(oss_output_object) }
snapshot_config = {'OutputFile': job_output}
```

- Time

```
snapshot_config['Time'] = 2
```

- Interval/Num

```
snapshot_config['Interval'] = 2
snapshot_config['Num'] = 3
```

- Width/Height

```
snapshot_config['Height'] = 360
```

4. Initiate API request and display returned value.

```
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print response
print 'RequestId is:', response['RequestId']
print 'JobId is:', response['SnapshotJob']['Id']
print 'http://%s.%s.aliyuncs.com/output_00001.jpg' % (oss_bucket, oss_location)
print 'http://%s.%s.aliyuncs.com/output_00002.jpg' % (oss_bucket, oss_location)
print 'http://%s.%s.aliyuncs.com/output_00003.jpg' % (oss_bucket, oss_location)
```

Complete code

```
# -*- coding: utf8 -*-
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitSnapshotJobRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output_{Count}.jpg'
# AcsClient
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# request
request = SubmitSnapshotJobRequest.SubmitSnapshotJobRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
            'Bucket': oss_bucket,
            'Object': quote(oss_input_object) }
request.set_Input(json.dumps(job_input))
```

```

# SnapshotConfig->OutputFile
job_output = {'Location': oss_location,
              'Bucket': oss_bucket,
              'Object': quote(oss_output_object) }
snapshot_config = {'OutputFile': job_output}
# SnapshotConfig->Time
snapshot_config['Time'] = 2
# SnapshotConfig->Interval/Num
snapshot_config['Interval'] = 2
snapshot_config['Num'] = 3
# SnapshotConfig->Width/Height
snapshot_config['Height'] = 360
# SnapshotConfig
request.set_SnapshotConfig(json.dumps(snapshot_config))
# PipelineId
request.set_PipelineId(pipeline_id)
# call api
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print response
print 'RequestId is:', response['RequestId']
print 'JobId is:', response['SnapshotJob']['Id']
print 'http://%s.%s.aliyuncs.com/output_00001.jpg' % (oss_bucket, oss_location)
print 'http://%s.%s.aliyuncs.com/output_00002.jpg' % (oss_bucket, oss_location)
print 'http://%s.%s.aliyuncs.com/output_00003.jpg' % (oss_bucket, oss_location)

```

3.4.7 Query media - Use OSS file address

If you do not know the media ID (a live video converted to an on-demand video using the media workflow), you can use the media input URL to query the media information over

QueryMediaListByURL.

```

import json
from aliyunsdkmts.request.v20140618 import QueryMediaListByURLRequest
from aliyunsdkcore import client
import urllib
region = '<region>'
access_key_id = '<accessKeyId>'
access_key_secret = '<accessKeySecret>'
def queryMediaListByURL():
    global client
    client = client.AcsClient(access_key_id, access_key_secret, region)
    request = QueryMediaListByURLRequest.QueryMediaListByURLRequest()
    ossDomain = 'http://<input-bucket>.<region>.aliyuncs.com/';
    #Encode ossObject
    ossObject = encodeByRFC3986("test/The Legend of the Swordsman.mp4")
    request.set_FileURLs(ossDomain + ossObject)
    response = client.do_action_with_exception(request);
    json_response = json.loads(response)
    print json_response
def encodeByRFC3986(ossObject):
    return urllib.quote(ossObject)
if __name__ == "__main__":

```

```
queryMediaListByURL()
```

3.4.8 Create HLS standard encryption workflow

Introduction

This document is an example of calling API to create HLS standard encryption workflow. For more information about creating HLS standard encryption workflow and playing encryption videos, see [HLS encryption and play](#). For more information about MPS SDK, see [Installation](#).

Example

```
import json
from aliyunsdkmts.request.v20140618 import AddMediaWorkflowRequest
from aliyunsdkcore import client
REGION_ID = '<region>'
ACCESS_KEY_ID = '<accessKeyId>'
ACCESS_KEY_SECRET = '<accessKeySecret>'
PIPELINE_ID = "<PipelineId>"
TEMPLATE_ID = "S00000001-100020" #Transcoding template ID, m3u8 template, set as
needed
OSS_LOCATION = "<OssLocation>"
INPUT_BUCKET = "<InputBucket>"
INPUT_PATH = "<InputPath>" ##Example: "HLS-Encryption"
OUTPUT_BUCKET = "<OutputBucket>"
ENCRYPTION_TYPE = "hls-aes-128"
HLS_KEY_URI = "<URI of decryption key>" #Example: http://decrypt.testdomain.com
ACT_START = "Act-Start"
ACT_ENCRYPTION = "Act-HLS-Encryption"
ACT_REPORT = "Act-Report"
def addMediaWorkflow():
    global client
    client = client.AcsClient(ACCESS_KEY_ID, ACCESS_KEY_SECRET, REGION_ID)
    request = AddMediaWorkflowRequest.AddMediaWorkflowRequest()
    request.set_Topology(buildWorkflowTopology())
    request.set_Name("HLS encryption workflow py")
    response = client.do_action_with_exception(request)
    print json.loads(response)
def buildWorkflowTopology():
    workflow = {}
    workflow["Activities"] = buildActivities()
    workflow["Dependencies"] = buildDependencies()
    print json.dumps(workflow)
    return json.dumps(workflow)
def buildActivities():
    activities = {}
    activities[ACT_START] = buildStartActivity()
    activities[ACT_ENCRYPTION] = buildTranscodeActivity()
    activities[ACT_REPORT] = buildReportActivity()
    return activities
def buildStartActivity():
    startActivity = {}
    startActivity["Name"] = ACT_START
    startActivity["Type"] = "Start"
    startActivity["Parameters"] = buildStartParameters()
    return startActivity
def buildStartParameters():
    startParameters = {}
    startParameters["PipelineId"] = PIPELINE_ID
    startParameters["InputFile"] = buildInputFile();
```

```

    return startParameters
def buildInputFile():
    inputFile = {}
    inputFile["Bucket"] = INPUT_BUCKET
    inputFile["Location"] = OSS_LOCATION
    inputFile["ObjectPrefix"] = INPUT_PATH
    return inputFile
def buildTranscodeActivity():
    transcodeActivity = {}
    transcodeActivity["Name"] = ACT_ENCRYPTION
    transcodeActivity["Type"] = "Transcode"
    transcodeActivity["Parameters"] = buildTranscodeParameters()
    return transcodeActivity
def buildTranscodeParameters():
    transcodeParameters = {}
    transcodeParameters["OutputBucket"] = OUTPUT_BUCKET
    transcodeParameters["OutputLocation"] = OSS_LOCATION
    transcodeParameters["Outputs"] = buildOutputsConfig()
    return transcodeParameters
def buildOutputsConfig():
    outputs = []
    output = {}
    output["ObjectRegex"] = ACT_ENCRYPTION +("/{RunId}/{FileName}"
    output["TemplateId"] = TEMPLATE_ID
    output["Encryption"] = buildEncryption()
    outputs.append(output)
    return outputs
def buildEncryption():
    encryption = {}
    encryption["Type"] = ENCRYPTION_TYPE
    encryption["KeyUri"] = HLS_KEY_URI
    return encryption
def buildReportActivity():
    reportActivity = {}
    reportActivity["Name"] = ACT_REPORT
    reportActivity["Parameters"] = buildReportParameters()
    reportActivity["Type"] = "Report"
    return reportActivity
def buildReportParameters():
    parameters = {}
    parameters["PublishType"] = "Auto"
    return parameters
def buildDependencies():
    dependencies = {}
    subActivityOfStart = [ACT_ENCRYPTION]
    dependencies[ACT_START] = subActivityOfStart
    subActivityOfTranscode = [ACT_REPORT]
    dependencies[ACT_ENCRYPTION] = subActivityOfTranscode
    dependencies[ACT_REPORT] = []
    return dependencies
if __name__ == "__main__":

```

```
addMediaWorkflow()
```

3.4.9 Splicing and simple cutting

1. Create AcsClient instance.

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. Create request and set parameters.

```
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
```

3. Sets transcode parameters.

- Input

```
job_input = {'Location': oss_location,
            'Bucket': oss_bucket,
            'Object': quote(head_object) }
request.set_Input(json.dumps(job_input))
```

- Output

```
output = {'OutputObject': quote(oss_output_object)}
```

- Video

```
output['Video'] = {'Width': 1280,
                  'Height': 720}
```

- MergeList

```
merge_video = {'MergeURL': 'http://%s.%s.aliyuncs.com/%s'%(oss_bucket,
oss_location, quote(oss_input_object))}
merge_tail = {'MergeURL': 'http://%s.%s.aliyuncs.com/%s'%(oss_bucket,
oss_location, quote(tail_object))}
output['MergeList'] = [merge_video, merge_tail]
```

4. Initiate API request and display returned value.

```
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'Requestid is:', response['Requestid']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']['JobId']
else:
    print ('SubmitJobs Failed code:',
          response['JobResultList']['JobResult'][0]['Code'],
          'message:',
          response['JobResultList']['JobResult'][0]['Message'])
```

Full codes

```
# -*- coding: utf8 -*-
import json
from urllib import quote
```

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitJobsRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
template_id = 'S00000001-200030'
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output.mp4'
head_object = 'head.mp4'
tail_object = 'tail.mp4'
# AcsClient
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# request
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
            'Bucket': oss_bucket,
            'Object': quote(head_object) }
request.set_Input(json.dumps(job_input))
# Output
output = {'OutputObject': quote(oss_output_object)}
# Output->TemplateId
output['TemplateId'] = template_id
# Output->Video
output['Video'] = {'Width': 1280,
                  'Height': 720}
# Output->MergeList
merge_video = {'MergeURL': 'http://%.%.aliyuncs.com/%%s'%(oss_bucket, oss_location,
quote(oss_input_object))}
merge_tail = {'MergeURL': 'http://%.%.aliyuncs.com/%%s'%(oss_bucket, oss_location,
quote(tail_object))}
output['MergeList'] = [merge_video, merge_tail]
# Outputs
outputs = [output]
request.set_Outputs(json.dumps(outputs))
request.set_OutputBucket(oss_bucket)
request.set_OutputLocation(oss_location)
# PipelineId
request.set_PipelineId(pipeline_id)
# call api
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']['JobId']
else:
    print ('SubmitJobs Failed code:',
          response['JobResultList']['JobResult'][0]['Code'],
          ' message:',
```

```
response['JobResultList']['JobResult'][0]['Message'])
```

3.4.10 Splicing-opening and ending scenes

1. Create AcsClient instance.

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. Create request and set parameters.

```
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
```

3. Sets transcode parameters.

- Input

```
job_input = {'Location': oss_location,
            'Bucket': oss_bucket,
            'Object': quote(head_object) }
request.set_Input(json.dumps(job_input))
```

- Output

```
output = {'OutputObject': quote(oss_output_object)}
```

- Video

```
output['Video'] = {'Width': 1280,
                  'Height': 720}
```

- OpeningList

```
opening_video = {'OpenUrl': 'http://%s.%s.aliyuncs.com/%s'%(oss_bucket,
oss_location, quote(head_object)),
                'Width': 640,
                'Start': 2}
output['OpeningList'] = [opening_video]
```

- TailSlateList

```
tailslate_video = {'TailUrl': 'http://%s.%s.aliyuncs.com/%s'%(oss_bucket,
oss_location, quote(tail_object)),
                  'Width': 640,
                  'BlendDuration': 3,
                  'BgColor': 'Black'}
output['TailSlateList'] = [tailslate_video]
```

4. Initiate API request and display returned value.

```
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']['JobId']
else:
    print ('SubmitJobs Failed code:',
```

```
response['JobResultList']['JobResult'][0]['Code'],
'message:',
response['JobResultList']['JobResult'][0]['Message'])
```

Full codes

```
# -*- coding: utf8 -*-
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitJobsRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
template_id = 'S00000001-200030'
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output.mp4'
head_object = 'head.mp4'
tail_object = 'tail.mp4'
# AcsClient
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# request
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
            'Bucket': oss_bucket,
            'Object': quote(oss_input_object) }
request.set_Input(json.dumps(job_input))
# Output
output = {'OutputObject': quote(oss_output_object)}
# Output->TemplateId
output['TemplateId'] = template_id
# Output->OpeningList
opening_video = {'OpenUrl': 'http://%.%.aliyuncs.com/%%s'%(oss_bucket, oss_location,
quote(head_object)),
                'Width': 640,
                'Start': 2}
output['OpeningList'] = [opening_video]
# Output->TailSlateList
tailslate_video = {'TailUrl': 'http://%.%.aliyuncs.com/%%s'%(oss_bucket, oss_location,
quote(tail_object)),
                  'Width': 640,
                  'BlendDuration': 3,
                  'BgColor': 'Black'}
output['TailSlateList'] = [tailslate_video]
# Outputs
outputs = [output]
request.set_Outputs(json.dumps(outputs))
request.set_OutputBucket(oss_bucket)
request.set_OutputLocation(oss_location)
# PipelineId
request.set_PipelineId(pipeline_id)
# call api
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']['JobId']
else:
```



```
print ('SubmitJobs Failed code:',
      response['JobResultList']['JobResult'][0]['Code'],
      'message:',
      response['JobResultList']['JobResult'][0]['Message'])
```

3.4.11 Add media

Add a video file to MEdia Files, and the user can specify workflow to be triggered to process this file:

```
import json
from aliyunsdkcore.acs_exception.exceptions import ServerException, ClientException
from aliyunsdkmts.request.v20140618 import AddMediaRequest
from aliyunsdkcore import client
import urllib
import thread
# Step 1 set region
REGION = "cn-shenzhen";
mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
# Step 2.set accesskey & keySecret
accessKeyId = "";
accessKeySecret = "";
cli = client.AcsClient(accessKeyId, accessKeySecret, REGION)
def addMeida():
    request = AddMediaRequest.AddMediaRequest()
    request.set_FileURL("http://mtb-sz-in.oss-cn-shenzhen.aliyuncs.com/media/r180-ABC.mp4")
    request.set_MediaWorkflowId("829bed0300994057a49e4f16de957e34")
    try:
        response = cli.do_action_with_exception(request)
        json_response = json.loads(response)
        print json.dumps(json_response)
    except ServerException, e:
        print e.get_error_code(), e.get_error_msg()
    except ClientException, e:
        print e.get_error_code(), e.get_error_msg()
def encodeByRFC3986(ossObject):
    return urllib.quote(ossObject)
if __name__ == "__main__":
    addMeida()
```

3.4.12 Add media workflow

The user can assemble activities provided by MPS, such as transcode activity and screenshot activity into a topology. The topology is as follows:

```
import json
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmts.request.v20140618 import AddMediaWorkflowRequest
from aliyunsdkcore import client
import urllib
import thread
# Step 1 set region
REGION = "cn-shenzhen";
OSS_REGION = "oss-cn-shenzhen";
mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
# Step 2.set accesskey & keySecret
accessKeyId = "";
accessKeySecret = "";
```

```

# Step 3.set mps transcoding queue id
PIPELINE_ID = "38bba54d524448be92d277caaa8da118";
cl = client.AcsClient(accessKeyId, accessKeySecret, REGION)
def addMeidaWorkflow():
    request = AddMediaWorkflowRequest.AddMediaWorkflowRequest()
    request.set_Name("Sequential-workflow");
    startActivity = {
        "Type": "Start",
        "Parameters": {
            "InputFile": {
                "Bucket": "mtb-sz-in",
                "Location": OSS_REGION,
                "ObjectPrefix": "media/"
            },
            "PipelineId": PIPELINE_ID
        }
    }
    transcodeActivity = {
        "Type": "Transcode",
        "Parameters": {
            "Outputs": [
                {
                    "OutputObject": encodeByRFC3986("transcode/{ObjectPrefix}/{FileName}.{
ExtName}"),
                    "TemplateId": "S00000001-000070"
                }
            ],
            "OutputLocation": OSS_REGION,
            "OutputBucket": "mtb-sz-out"
        }
    }
    reportActivity = {
        "Type": "Report",
        "Parameters": {
        }
    }
    topology = {
        "Activities": {
            "startNode": startActivity,
            "transcodingNode": transcodeActivity,
            "reportNode": reportActivity
        },
        "Dependencies": {
            "startNode": ["transcodingNode"],
            "transcodingNode": ["reportNode"],
            "reportNode": []
        }
    }
    request.set_Topology(topology)
    try:
        response = json.loads(cl.do_action_with_exception(request))
        print json.dumps(response)
    except ServerException, e:
        print e.get_error_code(), e.get_error_msg()
def encodeByRFC3986(ossObject):
    return urllib.quote(ossObject)
if __name__ == "__main__":

```

```
addMeidaWorkflow()
```

3.5 PHP SDK

3.5.1 Overview

MPS PHP SDK is based on Alibab Cloud PHP SDK. This article introduces the basic knowledge about Alibaba Cloud PHP SDK.

- [Alibaba Cloud SDK Developer Guide > PHP SDK > User Guide](#)
- [Alibaba Cloud PHP SDK GitHub](#)

After learning about the basic knowledge, you can perform MPS PHP SDK installation. For more information, see [MPS > SDK Reference > Transcoding SDKs > PHP SDK > Installation](#).

3.5.2 Installation

This article introduces the installation method of Alibaba Cloud PHP SDK.

1. Download the source code.

```
git clone https://github.com/aliyun/aliyun-openapi-php-sdk.git
```

2. Add the reference.

Assume the PHP SDK is downloaded to the path `/path/to/aliyun-openapi-php-sdk`.

```
require_once '/path/to/aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
```

Assume `aliyun-openapi-php-sdk` is referred to under the current directory of the project.

```
require_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
```

3. Automatically load MPS SDK.

Edit the file `aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php`.

Find the content `//config sdk auto load path.`, and add `Autoloader::addAutoloadPath("aliyun-php-sdk-mts");` at the end.

3.5.3 Quick start

1. Create AcsClient instance.

```
$clientProfile = DefaultProfile::getProfile(  
    $mps_region_id,          # Your Region ID  
    $access_key_id,         # Your AccessKey ID  
    $access_key_secret      # Your AccessKey Secret  
);
```

```
$client = new DefaultAcsClient($clientProfile);
```

2. Create request and set parameters.

```
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
```

3. Initiate API request and display returned value.

```
$response = $client->getAcsResponse($request);
print 'PipelineName is:' . $response->{'PipelineList'}->{'Pipeline'}[0]->{'Name'} . "\n";
print 'PipelineId is:' . $response->{'PipelineList'}->{'Pipeline'}[0]->{'Id'} . "\n";
```

Full codes

```
<? php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_ids = array('cn-hangzhou', 'cn-beijing', 'cn-shenzhen',
                        'cn-shanghai', 'cn-hongkong', 'us-west-1',
                        'ap-southeast-1', 'ap-northeast-1', 'eu-central-1',
                        'ap-south-1');
foreach ($mps_region_ids as $mps_region_id) {
    print 'region is:' . $mps_region_id . "\n";
    # Create DefaultAcsClient instance and complete initialization
    $clientProfile = DefaultProfile::getProfile(
        $mps_region_id,          # Your Region ID
        $access_key_id,         # Your AccessKey ID
        $access_key_secret      # Your AccessKey Secret
    );
    $client = new DefaultAcsClient($clientProfile);
    # Create API request and set parameters
    $request = new Mts\SearchPipelineRequest();
    # Initiate request and handle the returned value
    try {
        $response = $client->getAcsResponse($request);
        print 'PipelineName is:' . $response->{'PipelineList'}->{'Pipeline'}[0]->{'Name'} . "\n";
        print 'PipelineId is:' . $response->{'PipelineList'}->{'Pipeline'}[0]->{'Id'} . "\n";
    } catch (ServerException $e) {
        print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
    } catch (ClientException $e) {
        print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
    }
}
```

3.5.4 Transcoding

1. Create AcsClient instance.

```
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,          # Your Region ID
    $access_key_id,         # Your AccessKey ID
    $access_key_secret      # Your AccessKey Secret
);
```

```
$client = new DefaultAcsClient($clientProfile);
```

2. Create request, and set parameters.

```
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
```

3. Transcoding parameters.

- Input

```
$input = array('Location' => $oss_location,
              'Bucket' => $oss_bucket,
              'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
```

- Output

```
$output = array('OutputObject' => urlencode($oss_output_object));
```

- Container

```
$output['Container'] = array('Format' => 'mp4');
```

- Video

```
$output['Video'] = array('Codec' => 'H. 264',
                        'Bitrate' => 1500,
                        'Width' => 1280,
                        'Fps' => 25);
```

- Audio

```
$output['Audio'] = array('Codec' => 'AAC',
                        'Bitrate' => 128,
                        'Channels' => 2,
                        'Samplerate' => 44100);
```

- TemplateId

```
$output['TemplateId'] = $template_id;
```

- PipelineId

```
$request->setPipelineId($pipeline_id);
```

4. Initiate API request and display returned value.

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
print 'JobId is:' .
    $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}->{'JobId'} . "\n";
} else {
print 'SubmitJobs Failed code:' .
    $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .
    'message:' .
    $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} . "\n";
```

}

Full code

```

<? php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$template_id = 'S00000001-200010';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'xxx';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output.mp4';
# Create DefaultAcsClient instance and complete initialization
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,          # Your Region ID
    $access_key_id,        # Your AccessKey ID
    $access_key_secret     # Your AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# Create API request and set parameters
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
              'Bucket' => $oss_bucket,
              'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
# Output
$output = array('OutputObject' => urlencode($oss_output_object));
# Ouput->Container
$output['Container'] = array('Format' => 'mp4');
# Ouput->Video
$output['Video'] = array('Codec' => 'H. 264',
                        'Bitrate' => 1500,
                        'Width' => 1280,
                        'Fps' => 25);
# Ouput->Audio
$output['Audio'] = array('Codec' => 'AAC',
                        'Bitrate' => 128,
                        'Channels' => 2,
                        'Samplerate' => 44100);
# Ouput->TemplateId
$output['TemplateId'] = $template_id;
$outputs = array($output);
$request->setOutputs(json_encode($outputs));
$request->setOutputBucket($oss_bucket);
$request->setOutputLocation($oss_location);
# PipelineId
$request->setPipelineId($pipeline_id);
# Initiate request and handle returned value
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";
    if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
        print 'JobId is:' .
            $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}->{'JobId'} . "\n";
    } else {
        print 'SubmitJobs Failed code:' .
            $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .

```

```

        'message:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} . "\n";
    }
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
}

```

3.5.5 Watermarks

1. Create AcsClient instance.

```

$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,      # Region ID
    $access_key_id,     # AccessKey ID
    $access_key_secret  # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);

```

2. Create request and set parameters.

```

$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');

```

3. Watermark parameters.

- Image watermark

```

$image_watermark_input = array(
    'Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($image_watermark_object)
);
$image_watermark = array(
    'WaterMarkTemplateId' => $watermark_template_id,
    'Type' => 'Image',
    'InputFile' => $image_watermark_input,
    'ReferPos' => 'TopRight',
    'Width' => 0.05,
    'Dx' => 0,
    'Dy' => 0
);

```

- Text watermark

```

$text_config = array(
    'Content' => '5rWL6K+V5paH5a2X5rC05Y2w',
    'FontName' => 'SimSun',
    'FontSize' => 16,
    'FontColor' => 'Red',
    'FontAlpha' => 0.5,
    'Top' => 10,
    'Left' => 10
);
$text_watermark = array(
    'WaterMarkTemplateId' => $watermark_template_id,
    'Type' => 'Text',
    'TextWaterMark' => $text_config
);

```

```
);
```

- Video watermark

```
$video_watermark_input = array (
    'Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($video_watermark_object)
);
$video_watermark = array(
    'WaterMarkTemplateId' => $watermark_template_id,
    'Type' => 'Image',
    'InputFile' => $video_watermark_input,
    'ReferPos' => 'BottomLeft',
    'Height' => 240,
    'Dx' => 0,
    'Dy' => 0
);
```

4. Initiate API request and display returned value.

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
    print 'JobId is:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}->{'JobId'} . "\n";
} else {
    print 'SubmitJobs Failed code:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .
        ' message:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} . "\n";
}
```

Full codes

```
<? php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$watermark_template_id = 'xxx';
$template_id = 'S00000001-200030';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'presigned';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output.mp4';
$image_watermark_object = 'logo.png';
$video_watermark_object = 'logo.mov';
# DefaultAcsClient
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,          # Region ID
    $access_key_id,        # AccessKey ID
    $access_key_secret     # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# request
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
```



```

        'Bucket' => $oss_bucket,
        'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
# Output
$output = array('OutputObject' => urlencode($oss_output_object));
# Output->TemplateId
$output['TemplateId'] = $template_id;
## Image Watermark
$image_watermark_input = array(
    'Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($image_watermark_object)
);
$image_watermark = array(
    'WaterMarkTemplateId' => $watermark_template_id,
    'Type' => 'Image',
    'InputFile' => $image_watermark_input,
    'ReferPos' => 'TopRight',
    'Width' => 0.05,
    'Dx' => 0,
    'Dy' => 0
);
## Text Watermark
$text_config = array(
    'Content' => '5rWL6K+V5paH5a2X5rC05Y2w',
    'FontName' => 'SimSun',
    'FontSize' => 16,
    'FontColor' => 'Red',
    'FontAlpha' => 0.5,
    'Top' => 10,
    'Left' => 10
);
$text_watermark = array(
    'WaterMarkTemplateId' => $watermark_template_id,
    'Type' => 'Text',
    'TextWaterMark' => $text_config
);
## Video Watermark
$video_watermark_input = array (
    'Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($video_watermark_object)
);
$video_watermark = array(
    'WaterMarkTemplateId' => $watermark_template_id,
    'Type' => 'Image',
    'InputFile' => $video_watermark_input,
    'ReferPos' => 'BottomLeft',
    'Height' => 240,
    'Dx' => 0,
    'Dy' => 0
);
# Output->Watermarks
$watermarks = array($image_watermark, $text_watermark, $video_watermark);
$output['WaterMarks'] = $watermarks;
# Outputs
$outputs = array($output);
$request->setOutputs(json_encode($outputs));
$request->setOutputBucket($oss_bucket);
$request->setOutputLocation($oss_location);
# PipelineId
$request->setPipelineId($pipeline_id);
# call api
try {

```

```

$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
    print 'JobId is:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}->{'JobId'} . "\n";
} else {
    print 'SubmitJobs Failed code:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .
        'message:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} . "\n";
}
} catch (ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
} catch (ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
}
}

```

3.5.6 Screenshot

1. Create AcsClient instance.

```

$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,      # Region ID
    $access_key_id,     # AccessKey ID
    $access_key_secret  # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);

```

2. Create request, and set parameters.

```

$request = new Mts\SubmitSnapshotJobRequest();
$request->setAcceptFormat('JSON');

```

3. Set screenshot parameters.

- Input

```

$input = array('Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));

```

- SnapshotConfig

- OutputFile

```

$output = array('Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($oss_output_object));
$snapshot_config = array('OutputFile' => $output);

```

- Time

```

$snapshot_config['Time'] = 2;

```

- Interval/Num

```

$snapshot_config['Interval'] = 2;

```

```
$snapshot_config['Num'] = 3;
```

- Width/Height

```
$snapshot_config['Height'] = 360;
```

4. Initiate API request and display returned value.

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";
print 'JobId is:' . $response->{'SnapshotJob'}->{'Id'} . "\n";
print 'http://'.$oss_bucket.'!'.$oss_location.'.aliyuncs.com/output_00001.jpg' . "\n";
print 'http://'.$oss_bucket.'!'.$oss_location.'.aliyuncs.com/output_00002.jpg' . "\n";
print 'http://'.$oss_bucket.'!'.$oss_location.'.aliyuncs.com/output_00003.jpg' . "\n";
```

Full code

```
<? php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'xxx';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output_{Count}.jpg';
# DefaultAcsClient
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,          # Region ID
    $access_key_id,         # AccessKey ID
    $access_key_secret      # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# request
$request = new Mts\SubmitSnapshotJobRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
# SnapshotConfig->OutputFile
$output = array('Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($oss_output_object));
$snapshot_config = array('OutputFile' => $output);
# SnapshotConfig->Time
$snapshot_config['Time'] = 2;
# SnapshotConfig->Interval/Num
$snapshot_config['Interval'] = 2;
$snapshot_config['Num'] = 3;
# SnapshotConfig->Width/Height
$snapshot_config['Height'] = 360;
# SnapshotConfig
$request->setSnapshotConfig(json_encode($snapshot_config));
# PipelineId
$request->setPipelineId($pipeline_id);
# call api
try {
    $response = $client->getAcsResponse($request);
```

```

print 'RequestId is:' . $response->{'RequestId'} . "\n";
print 'JobId is:' . $response->{'SnapshotJob'}->{'Id'} . "\n";
print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/output_00001.jpg' . "\n";
print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/output_00002.jpg' . "\n";
print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/output_00003.jpg' . "\n";
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
}

```

3.5.7 Splicing and simple cutting

1. Create AcsClient instance.

```

$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,      # Region ID
    $access_key_id,     # AccessKey ID
    $access_key_secret  # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);

```

2. Create request, and set parameters.

```

$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');

```

3. Set transcoding parameters.

- Input

```

$input = array('Location' => $oss_location,
              'Bucket' => $oss_bucket,
              'Object' => urlencode($head_object));
$request->setInput(json_encode($input));

```

- Output

```

$output = array('OutputObject' => urlencode($oss_output_object));

```

- Video

```

$output['Video'] = array('Width' => 1280,
                       'Height' => 720);

```

- MergeList

```

$merge_video = array('MergeURL' => 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/'.urlencode($oss_input_object));
$merge_tail = array('MergeURL' => 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/'.urlencode($tail_object));
$output['MergeList'] = array($merge_video, $merge_tail);

```

4. Initiate API request and display returned value.

```

$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {

```

```

    print 'JobId is:' .
      $response->{'JobResultList'}->{'JobResult'}[0]->{'JobId'} . "\n";
  } else {
    print 'SubmitJobs Failed code:' .
      $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .
      ' message:' .
      $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} . "\n";
  }
}

```

Full codes

```

include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$template_id = 'S00000001-200030';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'xxx';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output.mp4';
$head_object = 'head.mp4';
$tail_object = 'tail.mp4';
// Create a DefaultAcsClient instance and initialize it;
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,          # Region ID
    $access_key_id,         # AccessKey ID
    $access_key_secret      # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# Create an API request and set parameters
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
              'Bucket' => $oss_bucket,
              'Object' => urlencode($head_object));
$request->setInput(json_encode($input));
# Output
$output = array('OutputObject' => urlencode($oss_output_object));
# Output->Video
$output['Video'] = array('Width' => 1280,
                       'Height' => 720);
# Output->TemplateId
$output['TemplateId'] = $template_id;
# Output->MergeList
$merge_video = array('MergeURL' => 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/'.urlencode($oss_input_object));
$merge_tail = array('MergeURL' => 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/'.urlencode($tail_object));
$output['MergeList'] = array($merge_video, $merge_tail);
# Outputs
$outputs = array($output);
$request->setOutputs(json_encode($outputs));
$request->setOutputBucket($oss_bucket);
$request->setOutputLocation($oss_location);
# PipelineId
$request->setPipelineId($pipeline_id);
# call api
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";
}

```

```

if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
    print 'JobId is:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}->{'JobId'} . "\n";
} else {
    print 'SubmitJobs Failed code:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .
        ' message:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} . "\n";
}
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
}
}

```

3.5.8 Splicing-opening and ending scenes

1. Create AcsClient instance.

```

$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,          # Region ID
    $access_key_id,         # AccessKey ID
    $access_key_secret      # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);

```

2. Create request, and set parameters.

```

$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');

```

3. Set transcoding parameters.

- Input

```

$input = array('Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($head_object));
$request->setInput(json_encode($input));

```

- Output

```

$output = array('OutputObject' => urlencode($oss_output_object));

```

- Video

```

$output['Video'] = array('Width' => 1280,
    'Height' => 720);

```

- OpeningList

```

$opening_video = array('OpenUrl' => 'http://'.$oss_bucket.'.'.$oss_location.'.
    aliyuncs.com/'.urlencode($head_object),
    'Width' => 640,
    'Start' => 2);

```

```
$output['OpeningList'] = array($opening_video);
```

- TailSlateList

```
$tailslate_video = array('TailUrl' => 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/'.urlencode($tail_object),
    'Width' => 640,
    'BlendDuration' => 3,
    'BgColor' => 'Black');
$output['TailSlateList'] = array($tailslate_video);
```

4. Initiate API request and display returned value.

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
    print 'JobId is:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}->{'JobId'} . "\n";
} else {
    print 'SubmitJobs Failed code:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .
        ' message:' .
        $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} . "\n";
}
```

Full codes

```
<? php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$template_id = 'S00000001-200030';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'xxx';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output.mp4';
$head_object = 'head.mp4';
$tail_object = 'tail.mp4';
# Create DefaultAcsClient instance and complete initialization
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,          # Region ID
    $access_key_id,        # AccessKey ID
    $access_key_secret     # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
#Create API request and set parameters
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
    'Bucket' => $oss_bucket,
    'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
# Output
$output = array('OutputObject' => urlencode($oss_output_object));
# Output->TemplateId
$output['TemplateId'] = $template_id;
# Output->OpeningList
```

```

$opening_video = array('OpenUrl' => 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/.'.urlencode($head_object),
    'Width' => 640,
    'Start' => 2);
$output['OpeningList'] = array($opening_video);
# Output->TailSlateList
$tailslate_video = array('TailUrl' => 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/.'.urlencode($tail_object),
    'Width' => 640,
    'BlendDuration' => 3,
    'BgColor' => 'Black');
$output['TailSlateList'] = array($tailslate_video);
# Outputs
$outputs = array($output);
$request->setOutputs(json_encode($outputs));
$request->setOutputBucket($oss_bucket);
$request->setOutputLocation($oss_location);
# PipelineId
$request->setPipelineId($pipeline_id);
# call api
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is: ' . $response->{'RequestId'} . "\n";
    if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
        print 'JobId is: ' .
            $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}->{'JobId'} . "\n";
    } else {
        print 'SubmitJobs Failed code: ' .
            $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .
            ' message: ' .
            $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} . "\n";
    }
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
}

```

3.5.9 Create HLS standard encryption workflow

Overview

The article is an example of calling API to create HLS standard encryption workflow. For more information about creating workflow, see [AddMediaWorkflow](#). For more information about MPS SDK, see [Installation](#).

Code example

```

<? php
include_once 'aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
date_default_timezone_set('PRC');
class HLSEncryptionWorkflowDemo {
    private $client;
    private $region = '<region>';
    private $accessKeyId = '<accessKeyId>';
    private $accessKeySecret = '<accessKeySecret>';
    private $pipelineId = "<PipelineId>";
    private $templateId = "S00000001-100020"; #transcoding template ID, m3u8 template
, set as needed

```



```

private $ossLocation = "<OssLocation>";
private $inputBucket = "<InputBucket>";
private $inputPath = "<InputPath>"; #Example: "HLS-Encryption"
private $outputBucket = "<OutputBucket>";
private $encryptionType = "hls-aes-128";
private $hlsKeyUri = "<decryption key URI>"; #Example: http://decrypt.testdomain.com
private $actStart = "Act-Start";
private $actEncryption = "Act-HLS-Encryption";
private $actReport = "Act-Report";
function __construct() {
    $profile = DefaultProfile::getProfile($this->region, $this->accessKeyId, $this->
accessKeySecret);
    $this->client = new DefaultAcsClient($profile);
}
function addMediaWorkflow() {
    $request = new Mts\AddMediaWorkflowRequest();
    $request->setName("HLS encryption workflow php");
    $request->setTopology($this->buildWorkflowTopology());
    $response = $this->client->getAcsResponse($request);
    echo json_encode($response);
}
function buildWorkflowTopology() {
    $activities = $this->buildActivities();
    $dependencies = $this->buildDependencies();
    $workflow = array(
        "Activities" => $activities,
        "Dependencies" => $dependencies
    );
    echo json_encode($workflow)."\n";
    return json_encode($workflow);
}
function buildActivities() {
    $activities = [
        $this->actStart => $this->buildStartActivity(),
        $this->actEncryption => $this->buildTranscodeActivity(),
        $this->actReport => $this->buildReportActivity()
    ];
    return $activities;
}
function buildStartActivity() {
    $startActivity = array(
        "Name" => $this->actStart,
        "Type" => "Start",
        "Parameters" => $this->buildStartParameters()
    );
    return $startActivity;
}
function buildStartParameters() {
    $startParameters = array(
        "PipelineId" => $this->pipelineId,
        "InputFile" => $this->buildInputFile()
    );
    return $startParameters;
}
function buildInputFile() {
    $inputFile = array(
        "Bucket" => $this->inputBucket,
        "Location" => $this->ossLocation,
        "ObjectPrefix" => $this->inputPath
    );
    return $inputFile;
}
function buildTranscodeActivity() {
    $transcodeParameters = array(

```

```
        "Name" => $this->actEncryption,
        "Type" => "Transcode",
        "Parameters" => $this->buildTranscodeParameters()
    );
    return $transcodeParameters;
}
function buildTranscodeParameters() {
    $transcodeParameters = array(
        "OutputBucket" => $this->outputBucket,
        "OutputLocation" => $this->ossLocation,
        "Outputs" => $this->buildOutputsConfig()
    );
    return $transcodeParameters;
}
function buildOutputsConfig() {
    $output = array(
        "ObjectRegex" => $this->actEncryption." /{RunId}/{FileName}",
        "TemplateId" => $this->templateId,
        "Encryption" => $this->buildEncryption()
    );
    $outputs = array($output);
    return $outputs;
}
function buildEncryption() {
    $encryption = array(
        "Type" => $this->encryptionType,
        "KeyUri" => $this->hlsKeyUri
    );
    return $encryption;
}
function buildReportActivity() {
    $reportActivity = array(
        "Name" => $this->actReport,
        "Parameters" => (object)[],
        "Type" => "Report"
    );
    return $reportActivity;
}
function buildDependencies() {
    $subActivityOfStart = array(
        $this->actEncryption
    );
    $subActivityOfTranscode = array(
        $this->actReport
    );
    $dependencies = array(
        $this->actStart => $subActivityOfStart,
        $this->actEncryption => $subActivityOfTranscode,
        $this->actReport => []
    );
    return $dependencies;
}
}
$demo = new HLEncryptionWorkflowDemo();
$demo->addMediaWorkflow();
```

? >

3.5.10 MPS queue management

The system automatically creates an MPS queue when you open the MPS service. You can also use interfaces to manage MPS queue (pipeline). For example, **SearchPipeline**, **QueryPipelineList**, **UpdatePipeline**.

Search MPS queue

You can use the **SearchPipeline** interface to search the MPS queue information.

```
$region = '<region>';
$accessKeyId = '<accessKeyId>';
$accessKeySecret = '<accessKeySecret>';
$profile = DefaultProfile::getProfile($region, $accessKeyId, $accessKeySecret);
$client = new DefaultAcsClient($profile);
$request = new Mts\SearchPipelineRequest();
// If an error occurs, it can throw ClientException or ServerException.
$response = $client->getAcsResponse($request);
$pipelines = $response->PipelineList->Pipeline;
foreach ($pipelines as $pipeline) {
    echo 'pipeline id:' . $pipeline->Id . ', name:' . $pipeline->Name . ', state:' . $pipeline->State . "\n";
}
```

Query MPS queue

If you have the pipelineId, you can use pipelineId to call **QueryPipelineList** interface to query MPS queue information.

```
$region = '<region>';
$accessKeyId = '<accessKeyId>';
$accessKeySecret = '<accessKeySecret>';
// The known pipeline ID, separated by comma (,)
$pipelineIds = '<pipelineIds>';
$profile = DefaultProfile::getProfile($region, $accessKeyId, $accessKeySecret);
$client = new DefaultAcsClient($profile);
$request = new Mts\QueryPipelineListRequest();
$request->setPipelineIds($pipelineIds);
// If an error occurs, it can throw ClientException or ServerException.
$response = $client->getAcsResponse($request);
$pipelines = $response->PipelineList->Pipeline;
foreach ($pipelines as $pipeline) {
    echo 'pipeline id:' . $pipeline->Id . ', name:' . $pipeline->Name . ', state:' . $pipeline->State . "\n";
}
```

Update MPS queue

Use the **UpdatePipeline** interface to update MPS queue information, including MPS queue name and status. The status includes Active and Paused.

```
$region = '<region>';
$accessKeyId = '<accessKeyId>';
$accessKeySecret = '<accessKeySecret>';
```

```

$profile = DefaultProfile::getProfile($region, $accessKeyId, $accessKeySecret);
$client = new DefaultAcsClient($profile);
$request = new Mts\SearchPipelineRequest();
// If an error occurs, it can throw ClientException or ServerException.
$response = $client->getAcsResponse($request);
$pipelines = $response->PipelineList->Pipeline;
$pipeline = $pipelines[0];
$request = new Mts\UpdatePipelineRequest();
$request->setPipelineId($pipeline->Id);
$request->setName($pipeline->Name);
$request->setState($pipeline->State == 'Paused' ? 'Active' : 'Paused');
$response = $client->getAcsResponse($request);
$pipeline = $response->Pipeline;
echo 'pipeline id:' . $pipeline->Id . ', name:' . $pipeline->Name . ', state:' . $pipeline->
State . "\n";

```

3.5.11 Query media - Use OSS file address

If you do not know the media ID (a live video converted to an on-demand video using the media workflow), you can use the media input URL to query the media information over

QueryMediaListByURL.

```

<? php
include_once 'aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
date_default_timezone_set('PRC');
class QueryMediaListByUrlDemo {
    private $client;
    private $region = '<region>';
    private $accessKeyId = '<accessKeyId>';
    private $accessKeySecret = '<accessKeySecret>';
    function __construct()
    {
        $profile = DefaultProfile::getProfile($this->region, $this->accessKeyId, $this->
accessKeySecret);
        $this->client = new DefaultAcsClient($profile);
    }
    function queryMediaListByUrl()
    {
        $request = new Mts\QueryMediaListByUrlRequest();
        $ossDomain = 'http://<input-bucket>.<region>.aliyuncs.com/';
        #ossObject must be RFC3986-encoded.
        $ossObject = $this->encodeByRFC3986('test/The Legend of the Swordsman.mp4');
        $request->setFileURLs($ossDomain.$ossObject);
        $response = $this->client->getAcsResponse($request);
        echo json_encode($response);
    }
    function encodeByRFC3986($arg_1)
    {
        $encodeOssObject="";
        $arraylist = explode("/", $arg_1);
        for($i = 0; $i < count($arraylist); $i++)
        {
            $tmp = rawurlencode($arraylist[$i]);
            $encodeOssObject = $encodeOssObject.$tmp;
            if ($i !== count($arraylist) - 1) {
                $encodeOssObject = $encodeOssObject."/";
            }
        }
        return $encodeOssObject;
    }
}

```

```

}
$demo = new QueryMediaListByUrlDemo();
$demo->queryMediaListByUrl();
? >

```

3.5.12 Add media

Add a video file to Media files:

```

<? php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
# Step 1 set region
$REGION = "cn-shenzhen";
$OSS_REGION = "oss-cn-shenzhen";
$mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
# Step 2.set accesskey & keySecret
$accessKeyId = "";
$accessKeySecret = "";
# Create DefaultAcsClient instance and perform initialization
$clientProfile = DefaultProfile::getProfile(
    $REGION,                # Your Region ID
    $accessKeyId,           # Your AccessKey ID
    $accessKeySecret        #Your AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
$request = new Mts\AddMediaRequest();
$request->setAcceptFormat('JSON');
$request->setFileURL("http://mtb-sz-in.oss-cn-shenzhen.aliyuncs.com/media/r180-ABC.mp4");
$request->setMediaWorkflowId("829bed0300994057a49e4f16de957e34");
# Initiate request and handle returned result
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is: ' . $response->{'RequestId'} . "\n";
    print "Response:".json_encode($response);
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
}
? >

```

3.5.13 Add media workflow

The user can assemble activities provided by MPS, such as transcode activity and screenshot activity into a topology. The topology is as follows:

```

<? php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
# Step 1 set region
$REGION = "cn-shenzhen";
$OSS_REGION = "oss-cn-shenzhen";
$mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
# Step 2.set accesskey & keySecret
$accessKeyId = "";
$accessKeySecret = "";
# Step 3.set mps transcoding queue id
$PIPELINE_ID = "38bba54d524448be92d277caaa8da118";

```

```

#Create DefaultAcClient instance and perform initialization
$clientProfile = DefaultProfile::getProfile(
    $REGION,          # Your Region ID
    $accessKeyId,    # Your AccessKey ID
    $accessKeySecret # Your AccessKey Secret
);
$client = new DefaultAcClient($clientProfile);
$request = new Mts\AddMediaWorkflowRequest();
$request->setAcceptFormat('JSON');
$request->setName("Sequential-workflow");
$startActivity = array(
    "Type"=>"Start",
    "Parameters"=>array(
        "InputFile"=>array(
            "Bucket"=> "mtb-sz-in",
            "Location"=> $OSS_REGION,
            "ObjectPrefix"=> "media/"
        ),
        "PipelineId"=>$PIPELINE_ID
    )
);
$transcodeActivity = array(
    "Type"=>"Transcode",
    "Parameters"=> array (
        "Outputs"=>array(
            array(
                "OutputObject"=> urlencode("transcode/{ObjectPrefix}/{FileName}.{ ExtName
}"),
                "TemplateId"=> "S00000001-000070"
            )
        ),
        "OutputLocation"=> $OSS_REGION,
        "OutputBucket"=>"mtb-sz-out"
    )
);
$reportActivity = array(
    "Type"=> "Report",
    "Parameters"=> array(
        "PublishType"=>"Auto"
    )
);
$topology = array(
    "Activities"=> array(
        "startNode"=>$startActivity,
        "transcodingNode"=>$transcodeActivity,
        "reportNode"=>$reportActivity
    ),
    "Dependencies"=>array (
        "startNode"=>array("transcodingNode"),
        "transcodingNode"=>array("reportNode"),
        "reportNode"=>array()
    )
);
$request->setTopology(json_encode($topology));
# Initiate request and handle returned result
try {
    $response = $client->getAcResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";
    print "Response:".json_encode($response);
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->getMessage() . "\n";
}
}

```

? >

4 Demo project

Currently, Alibaba Cloud provides the MPS SDKs for Java, Python, and PHP. For details about the installation and usage instructions, see the following URLs:

- [Multi-region support](#)
- [Java](#)
- [PHP](#)
- [Python](#)

[Demo project](#)