Media Processing Service

Development Guide

Product Documentation

Tencent Cloud

Copyright Notice

Trademark Notice

Tencent Cloud

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

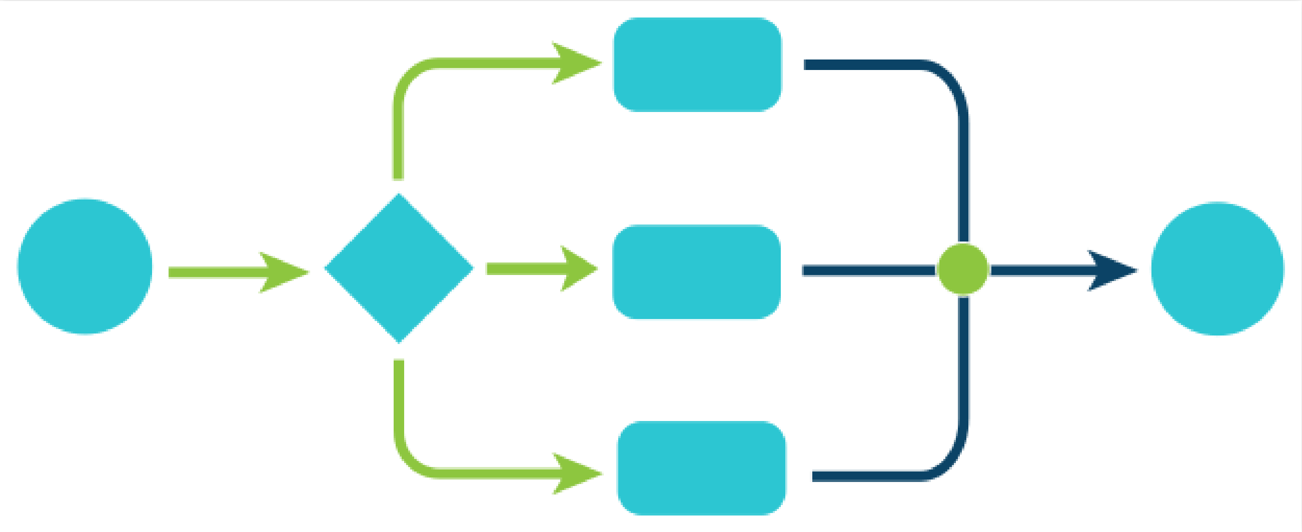
Development Guide

Concepts

Workflow

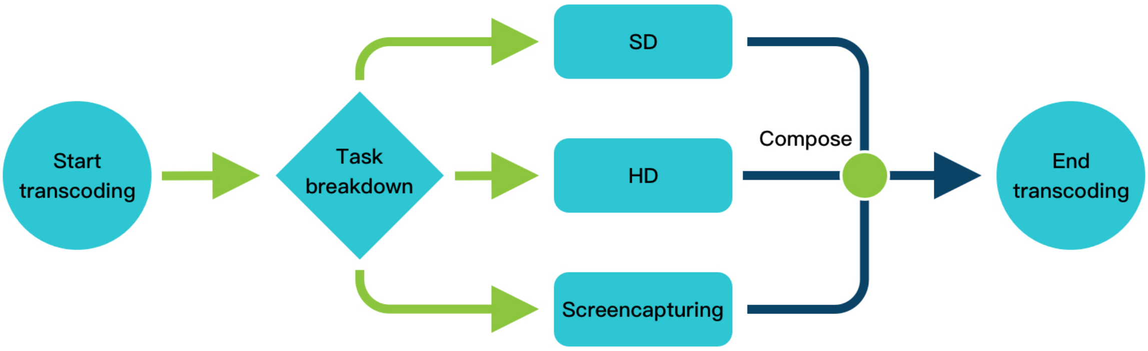
Last updated：2021-12-22 16:06:47

A workflow refers to a set of job tasks performed on a source audio/video file. Job tasks can be parallel or serial and are simply referred to as tasks in MPS. A workflow is as shown below.



Symbol	Meaning
Circle	Start and end of a task
Diamond	Task breakdown
Rectangle	Task unit
Dot	Assembly or combination of task units
Arrow	Order of executing different tasks or different steps of a task

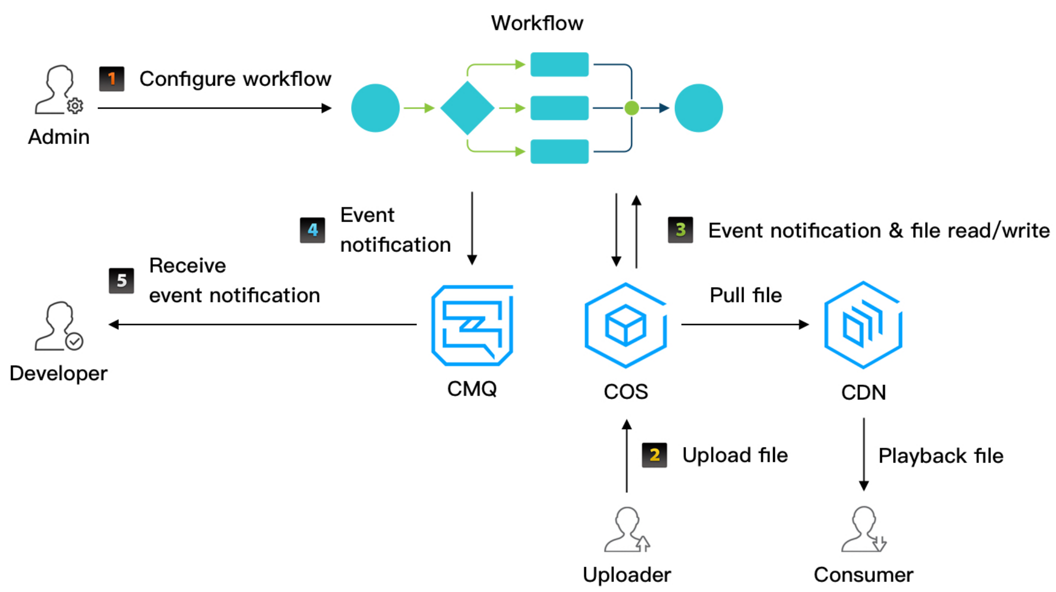
An MPS workflow may consist of tasks such as transcoding, sampled screenshot, time point screenshot, animated image generating, image sprite generating, and watermarking. Below is a typical example of an MPS workflow:



If a task involves multiple outputs, such as transcoding to SD and HD or taking screenshots of different sizes, it will be broken down into multiple subtasks that will be executed simultaneously. After all subtasks are completed, MPS will combine the results, and the task ends.

How a Workflow Works

The process of executing a workflow involves workflow configuration, task triggering, task execution, and event notification sending, as shown below:



1. **Workflow configuration**: You can configure a workflow in the console as an admin. Before configuration, you must create a CMQ queue and a COS bucket and grant your MPS service role access to the two services.
2. **Task triggering**: After an audio/video file is uploaded via the console or through an SDK to the COS bucket created, the task bound to the bucket will be triggered. You can also use the ProcessMedia API to initiate a task on a specific file.
3. **Task execution**: Read/Write operations such as downloading source files from COS and uploading output files to COS will be performed during task execution.
4. **Event notification sending**: After the workflow is completed, MPS will send a notification to the CMQ queue created. You can receive the notification through a CMQ API.

> Note：
>
> - For more information on workflows, see Workflow. For how to set up a workflow, see Setting Workflow.
> - After a file is transcoded successfully, you can proceed with your subsequent business logic, such as distributing the transcoded video through CDNs.

Configuring workflow

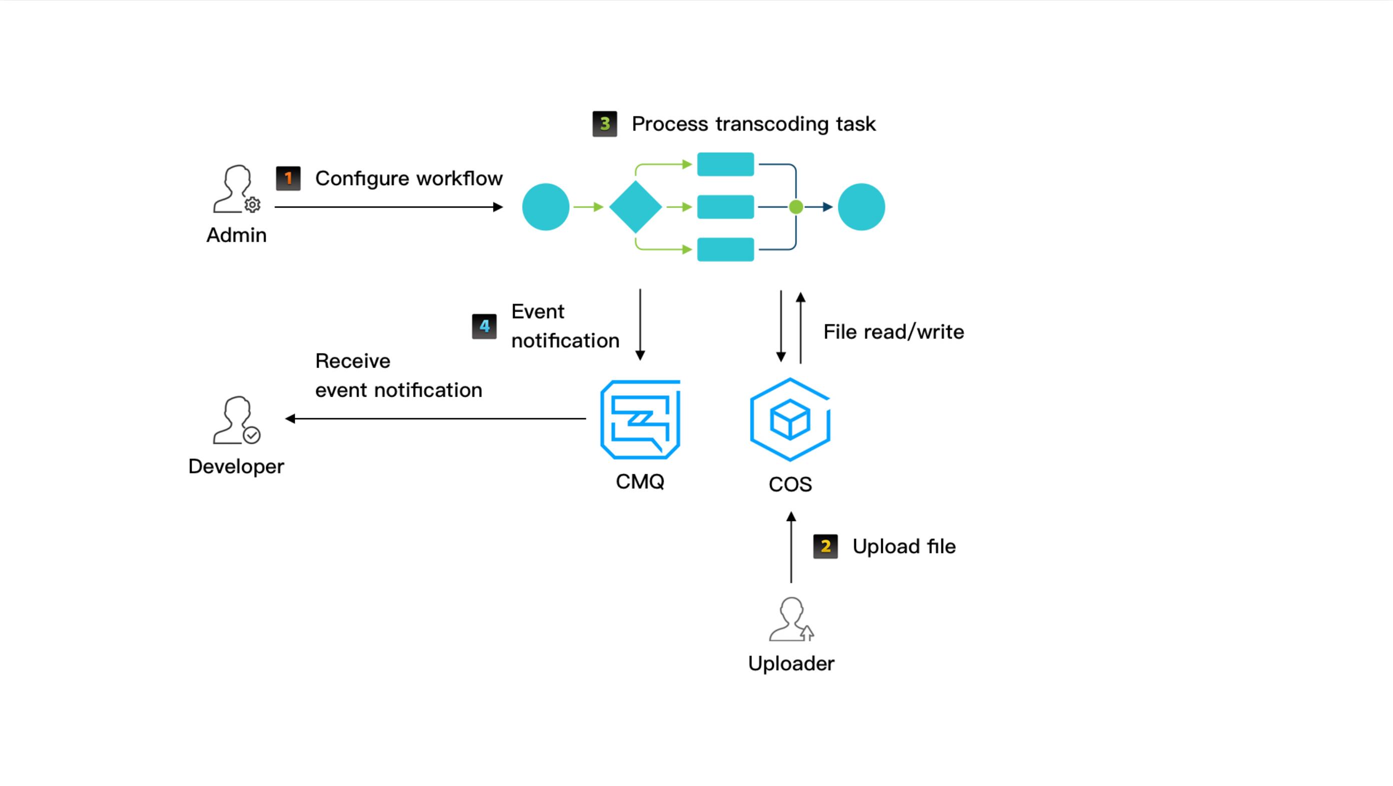
You need to configure a workflow if you want a task to be triggered automatically upon file upload. After configuration, the task will be initiated automatically on files uploaded to the specified COS bucket and the output files will be uploaded to the same or a different COS bucket.

You can also call an API to initiate a task on a specific file, in which case workflow configuration is not required.

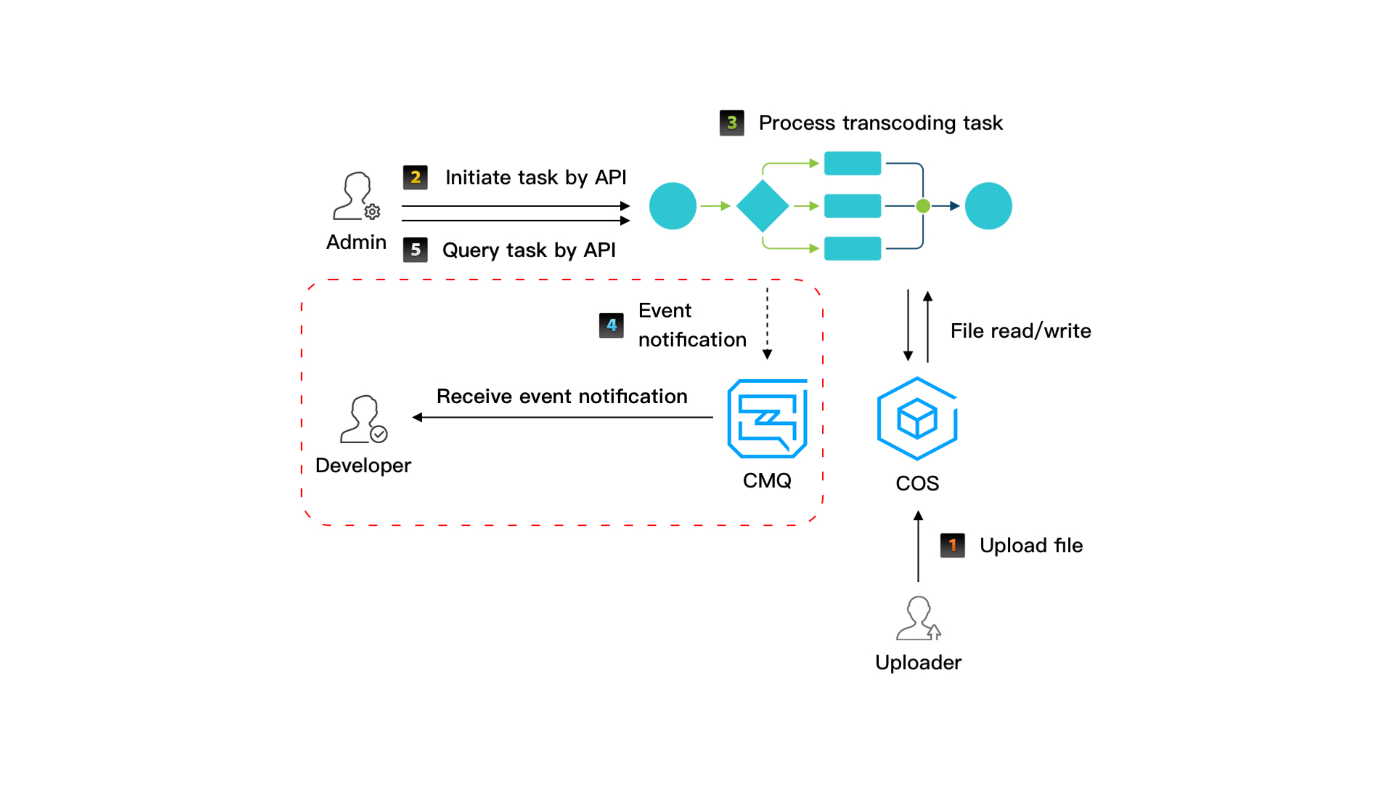
Triggering task

A transcoding task can be automatically or manually triggered.

- **Automatic triggering**: If a workflow is configured, the transcoding task will be automatically triggered upon file upload.



- **Manual triggering**: You can call an API to initiate a transcoding task and receive task completion notifications through CMQ or query the task status by `TaskId` . For details, see Manually Initiating Transcoding.



Note：

- Step 5 is the query of the task status via an API. The request parameter is the `TaskId` returned after task initiation.
- Steps in the red box are optional. When you manually trigger a transcoding task, you can use CMQ to receive notifications or query the task status through an API as shown in step 5.

Executing task

Task execution may involve transcoding, screenshot taking, watermarking, and output file upload. A task is broken down into several subtasks, which are executed in parallel or in series to speed up the process.

After the task is completed, MPS will upload the output files to the specified COS bucket. If upload fails, the task status will be "failed".

Sending event notification

You will be notified when a task is completed, whether it succeeds or fails. You can determine what to do next depending on the notification.

Template

Last updated：2024-06-12 14:52:03

To increase the ease of use, Media Processing Service (MPS) assembles commonly used key transcoding parameters into templates. A template is identified by its name and ID. For example, common templates named Fluent, SD, HD, and FHD are identified by IDs 10, 20, 30, and 40, respectively. There are different templates for different transcoding tasks:

Transcoding templates
Remuxing templates
Animated image templates
Time point screenshot templates
Sampled screenshot templates
Image sprite templates
Adaptive bitrate streaming templates
Intelligent auditing templates
Intelligent identification templates
Intelligent analysis templates

For the above template types, MPS provides corresponding commonly used templates, referred to as **system preset templates** . You can also create new parameter templates of various types and specify different parameter values for them, referred to as **custom parameter templates** . For detailed information on each parameter in the template, refer to Template Parameter Description.

System Preset Templates

The following provides information about system preset templates of various types, including template IDs and key parameter settings.

System Preset Transcoding Templates

Video

Clarity	Template ID	Format	Video Parameters				Audio Parameters		
			Resolution	Bitrate	Frame Rate	Codec	Bitrate	Sample Rate	Soun Chan
Smooth	100010	MP4	x 360	400 kbps	25	H.264	64 kbps	44,100 Hz	Stere
	100210	HLS							

SD	100020	MP4	x 540	1,000 kbps					
	100220	HLS							
HD	100030	MP4	x 720	1,800 kbps			128 kbps		
	100230	HLS							
FHD	100040	MP4	x 1080	2,500 kbps					
	100240	HLS							
2K	100070	MP4	x 1440	3,000 kbps			160 kbps		
	100270	HLS							
4K	100080	MP4	x 2160	6,000 kbps					
	100280	HLS							

Audio

Template ID	Format	Bitrate	Codec	Sound Channels	Sample Rate
1100	M4A	24kbps	AAC	Dual-channel	44100Hz
1110		48kbps			
1120		96kbps			
1130		192kbps			
1140		256kbps			
1010	MP3	128kbps	MP3		
1020		320kbps			

System Preset TSC Templates

Clarity	Template ID	Format	Video Parameters				Audio Parameters		
			Resolution	Maximum Bitrate	Frame Rate	Code	Bitrate	Sample Rate	Sc Ch
Same as	100800	MP4	Same as source	No limit	25	H.264	Same as	44,100 Hz	St

source							source		
Smooth	100810		x 360				64 kbps		
SD	100820		x 540						
HD	100830		x 720				128 kbps		
FHD	100840		x 1080						

System Preset Remuxing Templates

Template ID	Format
875	MP4
876	HLS

System Preset Animated Image Templates

Template ID	Format	Resolution	Frame Rate
20000	GIF	Same as source	2
20001	WebP	Same as source	2

System Preset Time Point Screenshot Templates

Template ID	Format	Width	Height	Fill Mode
10	JPG	Same as source	Same as source	Stretch

System Preset Sampled Screenshot Templates

Template ID &nbsp	Format	Width	Height	Interval Measurement	Interval	Fill Mode
10	JPG	Same as source	Same as source	By percent	10%	Scale to fill

System Preset Image Sprite Templates

Template ID	Format	Subimage Width	Subimage Height	Subimage Rows	Subimage Columns	Interval Measurement	Interval

10	JPG	142	80	10	10	By time	10s

System Preset Adaptive Bitrate Streaming Templates

Template information

Template ID	Package Type	Substream Info	Disable Low-Res to High-Res Conversion
10	HLS	Substreams for 6 clarity levels from "SD" to "4K"	Yes

Substream information

Substream Clarity	Video Parameters				Audio Parameters			
	Resolution	Bitrate	Frame Rate	Codec	Bitrate	Sample Rate	Sound Channels	Codec
Smooth	x 240	256 kbps	24	H.264	48 kbps	44,100 Hz	Stereo	AAC
SD	x 480	512 kbps	24	H.264	48 kbps	44,100 Hz	Stereo	AAC
HD	x 720	512 kbps	24	H.264	48 kbps	44,100 Hz	Stereo	AAC
FHD	x 1080	1,024 kbps	24	H.264	48 kbps	44,100 Hz	Stereo	AAC
2K	x 1440	3,072 kbps	24	H.264	48 kbps	44,100 Hz	Stereo	AAC
4K	x 2160	6,144 kbps	24	H.264	48 kbps	44,100 Hz	Stereo	AAC

System Preset Intelligent Auditing Templates

Template ID	Video Image			ASR		OCR	
	Porn	Terrorism	Politically Sensitive	Porn	Politically Sensitive	Porn	Politically Sensitive
10	Yes	Yes	Yes	No	No	No	No

20	Yes	Yes	Yes	Yes	Yes	Yes	Yes

System Preset Intelligent Identification Templates

Template ID	Full Text Recognition	Text Keyword Recognition	Full Speech Recognition	Spoken Keyword Recognition
10	No	No	No	No

System Preset Intelligent Analysis Templates

Template ID	Intelligent Classification	Intelligent Tagging	Intelligent Cover Generation	Intelligent Frame Tagging
10	Yes	Yes	Yes	No
20	Yes	Yes	Yes	Yes

Custom Parameter Templates

In addition to the system preset templates, you can also customize template parameters as needed, i.e., create a **custom template** . You can create templates of corresponding types through the console or by calling APIs. Such templates are only visible to yourself.

Creating a custom parameter template in the console

For how to create a custom template in the console, see Templates.

Creating a custom parameter template through an API

You can use the following APIs to create different types of custom parameter templates:

CreateTranscodeTemplate
CreateWatermarkTemplate
CreateSampleSnapshotTemplate
CreateSnapshotByTimeOffsetTemplate
CreateAnimatedGraphicsTemplate
CreateImageSpriteTemplate
CreateAdaptiveDynamicStreamingTemplate
CreateContentReviewTemplate
CreateAIRecognitionTemplate
CreateAIAnalysisTemplate

Uploading Video File

Video Upload

Last updated：2020-02-24 15:06:56

Video Upload Methods

MPS supports the following video upload methods:

- Upload via console: You can log in to the COS Console and upload local video files to a COS bucket. This method is suitable for uploading a small number of videos.
- Upload via client: You can upload local video files to a COS bucket through a COS SDK. This method features simple upload for small files and multipart upload for large ones. You can pause, resume or cancel uploads, making it suitable for both user generated content (UGC) and professionally generated content (PGC). The upload methods are as follows:
  - Simple Upload
  - Multipart Upload

Supported Audio/Video File Formats

- Video: MP4, TS, FLV, WMV, ASF, RM, RMVB, MPG, MPEG, 3GP, MOV, WEBM, MKV, and AVI
- Audio: MP3, M4A, FLAC, OGG, WAV, and AMR

> MPS will transcode files in the above container formats according to your workflow settings. Files not in the above formats will not be touched.

Triggering Transcoding Task

Setting Workflow

Last updated：2020-02-24 15:08:00

Complete the following steps to set a workflow:

- Set both trigger and output bucket and path
- Select the task items. Selections include transcoding, screenshot, animated image generating, and more.
- Set the CMQ address for event notification.

Created templates are used in task configuration. You can go to MPS Console > Template Settings for a complete list of parameter templates. For more information on how to create a custom parameter template, please see Parameter Templates.

Creating a Workflow

--Set-both-trigger-and-output-bucket-and-path">

- Set both trigger and output bucket and path

- Trigger bucket: the bucket where a source file is stored. Select a region and select the bucket located in the region.

- Trigger bucket directory: the directory that will trigger transcoding. The default value is the root directory, i.e., transcoding will be triggered for files in all directories in the bucket. **If this parameter is set, transcoding tasks will be initiated only for files in the specified directory.**

- Output bucket: the bucket where the output file will be stored. **Output buckets must be in the same region as the trigger bucket.**

- Output directory: the directory where the output file will be stored. The output path is the same as the source path by default. In this case, you need to ensure that the output file has a name different from that of the source file; otherwise, the source file will be overwritten. If this parameter is set, output files will be stored in the specified directory.

> If you set the output path in each task's configuration separately, the path settings in the task configuration will prevail.

Configuring event notification

MPS uses the queue model in CMQ as the event notification model. This is a messaging model where messages are written to one end and read from the other end. The writer is the producer, and the reader is the consumer.
In this model, each message can only be consumed once. In the simplest case, there are only one producer and one consumer. In complex cases, there can be multiple producers and multiple consumers, and each consumer will receive an equal portion of all messages.

> Only COS v4 regions are supported, i.e., Shanghai (`sh`), Guangzhou (`gz`), Chengdu (`cd`), Chongqing (`cq`), and Beijing (`bj`).

For more information on CMQ, please see CMQ Overview.

Configuring a transcoding task

Transcoding task configuration includes configuring tasks such as video transcoding, audio transcoding, and watermarks. You simply need to select the corresponding transcoding template.

Output path settings

Path Type	Description	Configuration Example	Output Example
Root directory	Ignores the set "output path"	Output directory: `/output/` Output path: `/transcode/{inputName}_{definition}.{format}`	Output file: `/transcode/testvideo_`
Relative directory	Utilizes the set "output path" as the directory prefix	Output directory: `/output/` Output path: `transcode/{inputName}_{definition}.{format}`	Output file: `/output/transcode/tes`

Configuring a screenshot task

Screenshot task configuration includes configuring tasks such as sampled screenshot, time point screenshot, and generating image sprite.

- Sampled screenshot: The system captures screenshots at a regular interval, which can be a fixed time or percent value, e.g., 1s or 1%.
- Time point screenshot: The system captures screenshots at specified time points, e.g., taking screenshots at 00:00:00, 00:05:00, and 00:15:30, respectively (three screenshots in total).
- Generating image sprite: The system combines multiple screenshots into a larger image. You can set the width and height of each screenshot.

Configuring an animated image generating task

Animated images are generated by taking screenshots of a video and combining them into an animated image in GIF or WEBP format. You can set the start and end time points for screencapturing.

Manually Initiating Transcoding

Last updated：2020-02-24 15:08:16

In MPS, there are two ways to initiate transcoding tasks:

- You can set up a workflow to automatically trigger a transcoding task upon file upload.
- You can call an API to manually initiate a transcoding task for an uploaded file.

This document describes how to call an API to initiate a task. To learn how to set up a workflow to automatically trigger transcoding tasks, see Setting Workflows.

Initiating a Transcoding Task

You can call the ProcessMedia API to initiate a transcoding task for a single file. If the API is successfully called, the task ID, i.e., the `TaskID` field in the result, will be returned.

Sample request

```
https://mps.tencentcloudapi.com/?Action=ProcessMedia
&InputInfo.Type=COS
&InputInfo.CosInputInfo.Bucket=TopRankVideo-125*****65
&InputInfo.CosInputInfo.Region=ap-chongqing
&InputInfo.CosInputInfo.Object=/movie/201907/WildAnimal.mov
&MediaProcessTask.TranscodeTaskSet.0.Definition=20
&MediaProcessTask.TranscodeTaskSet.1.Definition=30
&MediaProcessTask.TranscodeTaskSet.2.Definition=40
&<Common request parameter>
```

Sample response

```
{
"Response": {
"RequestId": "6ca31e3a-6b8e-4b4e-9256-fdc700064ef3",
"TaskId": "125****65-procedurev2-bffb15f07530b57bc1aabb01fac74bca"
}
}
```

If you have configured CMQ for event notification, you will receive a notification upon completion of this task. In addition to receiving event notifications through CMQ, you can also use the DescribeTaskDetail API to query the task result. The input parameter is `TaskId` returned by the ProcessMedia API.

Template Parameter Description

Last updated：2020-02-24 15:08:38

When creating a parameter template, you need to set the parameters, including bitrate, video width and height, remuxing format, and codec. This document describe the key parameters for each template and their value ranges.

Audio/Video Transcoding Template

Category	Parameter	Description
Muxing	Container format	The following video and audio container formats are supported: • Video: MP4, TS, HLS, and FLV • Audio: MP3, M4A, FLAC, and OGG
	Video stream deletion	If this parameter is enabled, the output video after transcoding will contain only the audio stream with the video stream discarded
	Audio stream deletion	If this parameter is enabled, the output video after transcoding will contain only the video stream with the audio stream discarded
Video encoding	Codec	H.264 and H.265 are supported
	Bitrate	Supported bitrate range: 10–35 Mbps
	Frame rate	Supported frame rate range: 1–60 fps; common values: 24, 25, and 30 fps
	Resolution	• Supported width range: 128–4,096 px • Supported height range: 128–4,096 px
	GOP length	Supported GOP length range: 1–10s
	Profile	• When the video codec is H.264, the `Baseline`, `Main`, and `High` profiles are supported • When the video codec is H.265, only the `Main` profile is supported
	Color space	YUV420P is supported
Audio encoding parameters	Codec	MP3, AAC, AC3, and FLAC are supported
	Sample rate	The following audio sample rates are supported: • 34,000 Hz • 44,100 Hz

		- 48,000 Hz
	Bitrate	Supported bitrate range: 26–256 Kbps, including the following values: - 48 Kbps - 64 Kbps - 128 Kbps
	Channel	- Mono - Dual - Stereo

Watermark Template

Parameter	Description
Type	Image and text watermarks are supported: - Image watermark: Static or animated images are supported - Text watermark: Texts in various languages are supported
Position	Relative position of a watermark in the video
Image Dimension	Size of an image watermark in the video
Image Content	Binary content of an image watermark
Font Size	Font size of a text watermark
Font Type	Font of a text watermark, e.g., Times New Roman
Font Color	Color of a text watermark, e.g., 0xRRGGBB
Font Alpha	Transparency of text watermark. Value range: 0–100%

Screenshot Template

Time point screenshot template

A time point screenshot template is used to take a screenshot at a specified time point or to generate a thumbnail cover.

Parameter	Description

Parameter	Description
Format	Output format of a screenshot file. Currently, only JPG is supported
Width	Screenshot width. Value range: 128–4,096 px
Height	Screenshot height. Value range: 128–4,096 px
Fill Type	Filling refers to the way of processing a screenshot when its aspect ratio is different from that of the source video. Generally, the following filling types are supported: • Stretch: The screenshot is stretched to match the aspect ratio of the source video, which may distort the image. • Fill in black: This option retains the aspect ratio of the source video for the screenshot and the unmatched area is filled in black. • Fill in white: This option retains the aspect ratio of the source video for the screenshot and the unmatched area is filled in white. • Gaussian blur: This option retains the aspect ratio of the source video for the screenshot and Gaussian blur is applied to the unmatched area.

Sampled screenshot template

A sampled screenshot template is used to take sampled screenshots.

Parameter	Description
Format	Output format of a screenshot file. Currently, only JPG is supported
Width	Screenshot width. Value range: 128–4,096 px
Height	Screenshot height. Value range: 128–4,096 px
Sample Type	The following two types are supported: • Sample by percent: If this is selected and `Interval` is set to `5%` for example, 20 screenshots will be generated • Sample by time: If this is selected and `Interval` is set to `10s` for example, the number of generated screenshots will depend on the video length
Interval	Sampling interval. • If the sampling type is by percent, this parameter will be a percent value • If the sampling type is by time, this parameter will be in seconds

Parameter	Description
Fill Type	Filling refers to the way of processing a screenshot when its aspect ratio is different from that of the source video. Generally, the following filling types are supported: • Stretch: The screenshot is stretched to match the aspect ratio of the source video, which may distort the image. • Fill in black: This option retains the aspect ratio of the source video for the screenshot and the unmatched area is filled in black. • Fill in white: This option retains the aspect ratio of the source video for the screenshot and the unmatched area is filled in white. • Gaussian blur: This option retains the aspect ratio of the source video for the screenshot and Gaussian blur is applied to the unmatched area.

Image sprite screenshot template

An image sprite screenshot template is used to take screenshots and combine them to generate an image sprite.

Parameter	Description
Format	Output format of an image sprite file. Currently, only JPG is supported
Width	Sub-image width
Height	Sub-image height
Rows	Number sub-image rows in an image sprite
Columns	Number sub-image columns in an image sprite
Sample Type	Sub-image sampling method. Currently, only sampling by time is supported
Interval	Time interval for capturing sub-images

- The value of `Width` * `Columns` should be between 128 and 4,096 px (i.e., the range of the image sprite width).
- The value of `Height` * `Rows` should be between 128 and 4,096 px (i.e., the range of the image sprite height).

Animated Image Generating Template

The target specification of an animated image is subject to parameters such as animated image format, width, height, and frame rate.

Parameter	Description
Format	Output format of an animated image file. Currently, only GIF and WEBP are supported
Width	Animated image width. Value range: 128–4,096 px
Height	Animated image height. Value range: 128–4,096 px
FPS	Supported frame rate range: 1–60 fps

Filename Variable

Last updated：2024-05-09 15:18:51

MPS supports rendering target paths of output files with the following variables:

Variable Name	Description
inputName	Input file name
inputFormat	Input file format
format	Output file format
definition	Parameter template ID
number	Output file number **Note：** This variable only takes effect on the output .ts files produced under the HLS format.

Sample 1

If your transcoding requirements are as follows:

The name of the input file is `AnimalWorldE01.mp4` .

Transcoding templates 100010, 100020, and 100030 are used.

The names of the output files are `AnimalWorldE01_100010.mp4` , `AnimalWorldE01_100020.mp4` , and `AnimalWorldE01_100030.mp4` , respectively.

Then, when using the ProcessMedia API to initiate transcoding:

You should specify the `InputInfo.CosInputInfo.OutputObjectPath` parameter as `{inputName}_{definition}.{format}` .

Sample 2

If your transcoding requirements are as follows:

The name of the input file is `AnimalWorldE01.mp4` .

Transcoding template 100210 is used.

The name of the output .m3u8 file is `AnimalWorldE01_from_mp4.m3u8` .

The names of the output .ts files are `AnimalWorldE01_from_mp4_0.ts` , `AnimalWorldE01_from_mp4_1.ts` , `AnimalWorldE01_from_mp4_2.ts` , and so on.

Then, when using the ProcessMedia API to initiate transcoding:

You should specify the `InputInfo.CosInputInfo.OutputObjectPath` parameter as `{inputName}_from_{inputFormat}.{format}` .

You should specify the `InputInfo.CosInputInfo.SegmentObjectName` parameter as `{inputName}_from_{inputFormat}_{number}.{format}` .

Querying Task

Last updated：2020-02-24 15:25:51

In addition to receiving notifications of file transcoding results through the event notification mechanism, MPS users can also query details of a specified transcoding task through the DescribeTaskDetail API. This API is generally used to query the progress and results of a transcoding task that was manually initiated by the ProcessMedia API. Returned results can be `WAITING` , `PROCESSING` , or `FINISH` .

- **WAITING**: The task has been initiated and is waiting to be processed.
- **PROCESSING**: The task is being processed.
- **FINISH**: The task has been completed.

Below are some task status samples:

`PROCESSING` sample

```
{
"Response":{
"TaskType":"WorkflowTask",
"Status":"PROCESSING",
"CreateTime":"2019-08-08T07:47:08Z",
"BeginProcessTime":"2019-08-08T07:47:09Z",
"FinishTime":"0000-00-00T00:00:00Z",
"WorkflowTask":{
"TaskId":"2451******-WorkflowTask-fc2172f5******a2e507cece0cb06fbet0",
"Status":"PROCESSING",
"ErrCode":0,
"Message":"",
"InputInfo":{
"Type":"COS",
"CosInputInfo":{
"Bucket":"macvc-1251132654",
"Region":"ap-chengdu",
"Object":"/abvc/111/2222/15692847.mp4"
}
},
"MetaData":{
"AudioDuration":204.2779998779297,
"AudioStreamSet":[
{
"Bitrate":127999,
"Codec":"mp3",
"SamplingRate":44100
```

```
}
],
"Bitrate":1232376,
"Container":"mov,mp4,m4a,3gp,3g2,mj2",
"Duration":204.2919921875,
"Height":720,
"Rotate":0,
"Size":31647438,
"VideoDuration":204.2919921875,
"VideoStreamSet":[
{
"Bitrate":1104377,
"Codec":"h264",
"Fps":24,
"Height":720,
"Width":1280
}
],
"Width":1280
},
"MediaProcessResultSet":[
{
"Type":"Transcode",
"TranscodeTask":{
"Status":"PROCESSING",
"ErrCode":0,
"Message":"SUCCESS",
"Input":{
"Definition":10,
"WatermarkSet":[

],
"OutputStorage":{
"Type":"COS",
"CosOutputStorage":{
"Bucket":"macyin**-12511*****",
"Region":"ap-beijing"
}
},
"OutputObjectPath":"/15692847_transcode_10",
"SegmentObjectName":"/15692847_transcode_10_{number}",
"ObjectNumberFormat":{
"InitialValue":0,
"Increment":1,
"MinLength":1,
"PlaceHolder":"0"
}
},
"Output":null
},
"AnimatedGraphicTask":null,
"SnapshotByTimeOffsetTask":null,
"SampleSnapshotTask":null,
"ImageSpriteTask":null
}
]
},
"TaskNotifyConfig":{
"CmqModel":"Queue",
"CmqRegion":"gz",
"QueueName":"macvtstest",
"TopicName":"",
"NotifyMode":"Change"
},
"TasksPriority":10,
"SessionId":"100",
"SessionContext":"100",
"RequestId":"13499555-145a-47f5-b6f6-64e829ed3b20"
}
}
```

`FINISH` sample

```
{
"Response": {
"TaskType": "WorkflowTask",
"Status": "FINISH",
"CreateTime": "2019-07-16T06:21:27Z",
"BeginProcessTime": "2019-07-16T06:21:28Z",
"FinishTime": "2019-07-16T06:21:46Z",
"WorkflowTask": {
"TaskId": "235303****-WorkflowTask-80108cc3380155d98b2e3573a48a******",
"Status": "FINISH",
"ErrCode": 0,
"Message": "",
"InputInfo": {
"Type": "COS",
"CosInputInfo": {
"Bucket": "vodtestbj-235303****",
"Region": "ap-beijing",
"Object": "/input/videoplayback.mp4"
}
},
"MetaData": {
```

```
"AudioDuration": 380.9465637207031,
"AudioStreamSet": [
{
"Bitrate": 95999,
"Codec": "aac",
"SamplingRate": 44100
}
],
"Bitrate": 409657,
"Container": "mov,mp4,m4a,3gp,3g2,mj2",
"Duration": 380.9465637207031,
"Height": 360,
"Rotate": 0,
"Size": 19626862,
"VideoDuration": 380.8804931640625,
"VideoStreamSet": [
{
"Bitrate": 313658,
"Codec": "h264",
"Fps": 29,
"Height": 360,
"Width": 480
}
],
"Width": 480
},
"MediaProcessResultSet": [
{
"Type": "Transcode",
"TranscodeTask": {
"Status": "SUCCESS",
"ErrCode": 0,
"Message": "SUCCESS",
"Input": {
"Definition": 210,
"WatermarkSet": [],
"OutputStorage": {
"Type": "COS",
"CosOutputStorage": {
"Bucket": "vodtestgz-235303****",
"Region": "ap-guangzhou"
}
},
"OutputObjectPath": "/output/{inputName}_transcode_{definition}.{format}",
"SegmentObjectName": "/output/{inputName}_transcode_{definition}_{number}",
"ObjectNumberFormat": {
"InitialValue": 0,
"Increment": 1,
"MinLength": 1,
"PlaceHolder": ""
}
},
"Output": {
"OutputStorage": {
"Type": "COS",
"CosOutputStorage": {
"Bucket": "vodtestgz-235303****",
"Region": "ap-guangzhou"
}
},
"Path": "/output/videoplayback_transcode_210.m3u8",
"Definition": 210,
"Bitrate": 353297,
"Height": 240,
"Width": 320,
"Size": 5692,
"Duration": 380.9580078125,
"Container": "hls,applehttp",
"Md5": "ae0dfe7c7336291d6243463b7bb14fea",
"VideoStreamSet": [
{
"Bitrate": 302307,
"Codec": "h264",
"Fps": 24,
"Height": 240,
"Width": 320
}
],
"AudioStreamSet": [
{
"Bitrate": 50990,
"Codec": "aac",
"SamplingRate": 44100
}
]
}
},
"AnimatedGraphicTask": null,
"SnapshotByTimeOffsetTask": null,
"SampleSnapshotTask": null,
"ImageSpriteTask": null
}
]
},
"TaskNotifyConfig": null,
"TasksPriority": 0,
"SessionId": "",
"SessionContext": "",
"RequestId": "requestId"
}
}
```

.

Event Notification Overview

Last updated：2022-09-28 15:59:53

An event notification informs you of a file transcoding result, so you can move on to the next logical step.

Event Notification Definition

An event in MPS refers to a task status change of a file during transcoding. An event notification refers to a message notification that you will receive at the end of an event, which includes the file transcoding result.

Event Notification Types

Below are types of event notifications currently provided:

Event Type	Event Name	Description
WorkflowTask	WorkflowTaskEvent	Status change. The `Status` field of the event shows the specific status. Generally, the status is `FINISH` , indicating that the task has ended. The task could have been completed successfully or failed.

Event Notification Mode

MPS uses TDMQ CMQ to send event notifications. When you use MPS, you need to activate TDMQ CMQ and authorize MPS before you can receive transcoding event notifications. If you do not activate TDMQ CMQ or do not configure the TDMQ CMQ queue address for event notification in a workflow template, you will not receive task event notifications from MPS.

Note：

- If you use a TencentCloud API to receive TDMQ CMQ event messages, you need to acknowledge each individual message for the message to be removed from the queue; otherwise, the API will keep pulling the same event messages.
- For more information on receiving TDMQ CMQ messages, please see Consuming Messages. For more information on acknowledging TDMQ CMQ messages, please see Deleting a Message.

Event Notification Sample

```
{
"EventType":"WorkflowTask",
"WorkflowTaskEvent":{
"TaskId":"245****654-WorkflowTask-f46dac7fe2436c47******d71946986t0",
"Status":"FINISH",
"ErrCode":0,
"Message":"",
"InputInfo":{
"Type":"COS",
"CosInputInfo":{
"Bucket":"macgzptest-125****654",
"Region":"ap-guangzhou",
"Object":"/dianping2.mp4"
}
},
"MetaData":{
"AudioDuration":11.261677742004395,
"AudioStreamSet":[
{
"Bitrate":127771,
"Codec":"aac",
"SamplingRate":44100
}
],
"Bitrate":2681468,
"Container":"mov,mp4,m4a,3gp,3g2,mj2",
"Duration":11.261677742004395,
"Height":720,
"Rotate":90,
"Size":3539987,
"VideoDuration":10.510889053344727,
"VideoStreamSet":[
{
"Bitrate":2553697,
"Codec":"h264",
"Fps":29,
"Height":720,
"Width":1280
}
],
"Width":1280
},
"MediaProcessResultSet":[
{
"Type":"Transcode",
"TranscodeTask":{
"Status":"SUCCESS",
"ErrCode":0,
"Message":"SUCCESS",
"Input":{
"Definition":10,
"WatermarkSet":[
{
"Definition":515247,
"TextContent":"",
"SvgContent":""
}
],
"OutputStorage":{
"Type":"COS",
"CosOutputStorage":{
"Bucket":"gztest-125****654",
"Region":"ap-guangzhou"
}
},
"OutputObjectPath":"/dasda/dianping2_transcode_10",
"SegmentObjectName":"/dasda/dianping2_transcode_10_{number}",
"ObjectNumberFormat":{
"InitialValue":0,
"Increment":1,
"MinLength":1,
"PlaceHolder":"0"
}
},
"Output":{
"OutputStorage":{
"Type":"COS",
"CosOutputStorage":{
"Bucket":"gztest-125****654",
"Region":"ap-guangzhou"
}
},
"Path":"/dasda/dianping2_transcode_10.mp4",
"Definition":10,
"Bitrate":293022,
"Height":320,
"Width":180,
"Size":401637,
"Duration":11.26200008392334,
"Container":"mov,mp4,m4a,3gp,3g2,mj2",
"Md5":"31dcf904c03d0cd78346a12c25c0acc9",
"VideoStreamSet":[
{
"Bitrate":244608,
"Codec":"h264",
"Fps":24,
"Height":320,
"Width":180
}
],
"AudioStreamSet":[
{
"Bitrate":48414,
"Codec":"aac",
"SamplingRate":44100
}
]
}
},
"AnimatedGraphicTask":null,
"SnapshotByTimeOffsetTask":null,
"SampleSnapshotTask":null,
"ImageSpriteTask":null
},
{
"Type":"AnimatedGraphics",
"TranscodeTask":null,
"AnimatedGraphicTask":{
"Status":"FAIL",
"ErrCode":30010,
"Message":"TencentVodPlatErr Or Unkown",
"Input":{
"Definition":20000,
"StartTimeOffset":0,
"EndTimeOffset":600,
"OutputStorage":{
"Type":"COS",
"CosOutputStorage":{
"Bucket":"gztest-125****654",
"Region":"ap-guangzhou"
}
},
"OutputObjectPath":"/dasda/dianping2_animatedGraphic_20000"
},
"Output":null
},
"SnapshotByTimeOffsetTask":null,
"SampleSnapshotTask":null,
"ImageSpriteTask":null
},
{
"Type":"SnapshotByTimeOffset",
"TranscodeTask":null,
"AnimatedGraphicTask":null,
"SnapshotByTimeOffsetTask":{
"Status":"SUCCESS",
"ErrCode":0,
"Message":"SUCCESS",
"Input":{
"Definition":10,
"TimeOffsetSet":[
],
"WatermarkSet":[
{
"Definition":515247,
"TextContent":"",
"SvgContent":""
}
],
"OutputStorage":{
"Type":"COS",
"CosOutputStorage":{
"Bucket":"gztest-125****654",
"Region":"ap-guangzhou"
}
},
"OutputObjectPath":"/dasda/dianping2_snapshotByOffset_10_{number}",
"ObjectNumberFormat":{
"InitialValue":0,
"Increment":1,
"MinLength":1,
"PlaceHolder":"0"
}
},
"Output":{
"Storage":{
"Type":"COS",
"CosOutputStorage":{
"Bucket":"gztest-125****654",
"Region":"ap-guangzhou"
}
},
"Definition":0,
"PicInfoSet":[
{
"TimeOffset":0,
"Path":"/dasda/dianping2_snapshotByOffset_10_0.jpg",
"WaterMarkDefinition":[
515247
]
}
]
}
},
"SampleSnapshotTask":null,
"ImageSpriteTask":null
},
{
"Type":"ImageSprites",
"TranscodeTask":null,
"AnimatedGraphicTask":null,
"SnapshotByTimeOffsetTask":null,
"SampleSnapshotTask":null,
"ImageSpriteTask":{
"Status":"SUCCESS",
"ErrCode":0,
"Message":"SUCCESS",
"Input":{
"Definition":10,
"OutputStorage":{
"Type":"COS",
"CosOutputStorage":{
"Bucket":"gztest-125****654",
"Region":"ap-guangzhou"
}
},
"OutputObjectPath":"/dasda/dianping2_imageSprite_10_{number}",
"WebVttObjectName":"/dasda/dianping2_imageSprite_10",
"ObjectNumberFormat":{
"InitialValue":0,
"Increment":1,
"MinLength":1,
"PlaceHolder":"0"
}
},
"Output":{
"Storage":{
"Type":"COS",
"CosOutputStorage":{
"Bucket":"gztest-125****654",
"Region":"ap-guangzhou"
}
},
"Definition":10,
"Height":80,
"Width":142,
"TotalCount":2,
"ImagePathSet":[
"/dasda/imageSprite/dianping2_imageSprite_10_0.jpg"
],
"WebVttPath":"/dasda/imageSprite/dianping2_imageSprite_10.vtt"
}
}
}
]
}
}
```

Video AI

Intelligent Video Recognition

Last updated：2022-01-18 16:17:11

MPS leverages AI technologies to recognize video content. The result of an intelligent video recognition task includes a recognition score, suggestion, and suspicious video segments. You can decide whether to expose a video based on the suggestion.

Result

MPS can intelligently recognize video images, speech (ASR), and optical characters (OCR).

Object	Operation	Description
Video images (People and objects)	Pornographic content	Checks for pornographic content in video images, including: • \`vulgar\`: vulgarity • \`intimacy\`: intimacy • \`sexy\`: sexiness
	Politically sensitive content	Checks for politically sensitive content in video images, including: • \`bloody\`: bloodiness • \`explosion\`: explosions and fires • \`violation_photo\`: banned icons • \`guns\`: weapons and guns
Speech (Speech to text)	Pornographic content	Checks for keywords for pornographic content in speech
	Politically sensitive content	Checks for keywords for politically sensitive content in speech
Optical characters Image to text	Pornographic content	Checks for keywords for pornographic content in images
	Politically sensitive content	Checks for keywords for politically sensitive content in images

Parameter description

Field Name	Type	Description

Field Name	Type	Description
confidence	Float	Intelligent recognition score (0-100). The higher the score, the more suspicious the content.
suggestion	String	There are three types of intelligent recognition suggestions: • `pass` : The degree of suspicion is not high, and approval is recommended. • `review` : The degree of suspicion is high, and human review is recommended. • `block` : The degree of suspicion is very high, and blocking is recommended.
segments	Array	Suspicious video segments, which help you locate suspicious content in a video

Initiating Task

Directions

You can call an API to initiate an intelligent video recognition task or configure automatic triggering of the task upon video upload.

- **API**: Call the ProcessMedia API, setting `AiContentReviewTask` to the ID of your intelligent video recognition template.
- **Automatic triggering upon upload**: In the console, create a workflow with intelligent video recognition enabled and upload videos to the bucket bound to the workflow.

Creating template

MPS uses templates to represent combinations of intelligent video recognition parameters, which determine which of the following operations MPS performs.

- Recognition of pornographic content in video images
- Recognition of politically sensitive content in video images
- Recognition of pornographic keywords in speech (ASR)
- Recognition of politically sensitive keywords in speech (ASR)
- Recognition of pornographic keywords in images (OCR)
- Recognition of politically sensitive keywords in images (OCR)

MPS provides preset intelligent video recognition templates for common parameter combinations. You can also use a server API to create and manage custom templates.

Obtaining Result

After initiating an intelligent video recognition task, you can wait for the result notification asynchronously or query the result synchronously.

Below is an example of the result returned after query (fields with null values are omitted):

```
{
"TaskType":"WorkflowTask",
"Status":"FINISH",
"CreateTime":"2019-07-16T06:21:27Z",
"BeginProcessTime":"2019-07-16T06:21:28Z",
"FinishTime":"2019-07-16T06:21:46Z",
"WorkflowTask":{
"TaskId":"2356768367-WorkflowTask-2e1af2456351812be963e309cc133403t0",
"Status":"FINISH",
"InputInfo":{
"Type":"COS",
"CosInputInfo":{
"Bucket":"MyVideoBucket-235303****",
"Region":"ap-beijing",
"Object":"/input/AnimalWorld.mp4"
}
},
"MetaData":{
"AudioDuration":60,
"AudioStreamSet":[
{
"Bitrate":383854,
"Codec":"aac",
"SamplingRate":48000
}
],
"Bitrate":1021028,
"Container":"mov,mp4,m4a,3gp,3g2,mj2",
"Duration":60,
"Height":480,
"Rotate":0,
"Size":7700180,
"VideoDuration":60,
"VideoStreamSet":[
{
"Bitrate":637174,
"Codec":"h264",
"Fps":23,
"Height":480,
"Width":640
}
],
"Width":640
},
"MediaProcessResultSet":[
],
"AiContentReviewResultSet":[
{
"Type":"Porn",
"PornTask":{
"Status":"SUCCESS",
"ErrCode":0,
"Message":"",
"Input":{
"Definition":10
},
"Output":{
"Confidence":98,
"Suggestion":"block",
"Label":"sexy",
"SegmentSet":[
{
"StartTimeOffset":9.5,
"EndTimeOffset":14,
"Confidence":98,
"Suggestion":"block",
"Label":"sexy",
"Url":"http://xxx.vod2.myqcloud.com/xxx/xxx/xx1.jpg",
"PicUrlExpireTime":"2019-07-23T06:21:46Z"
},
{
"StartTimeOffset":16.5,
"EndTimeOffset":18,
"Confidence":80,
"Suggestion":"review",
"Label":"sexy",
"Url":"http://xxx.vod2.myqcloud.com/xxx/xxx/xx2.jpg",
"PicUrlExpireTime":"2019-07-23T06:21:46Z"
},
{
"StartTimeOffset":41,
"EndTimeOffset":49,
"Confidence":97,
"Suggestion":"block",
"Label":"sexy",
"Url":"http://xxx.vod2.myqcloud.com/xxx/xxx/xx3.jpg",
"PicUrlExpireTime":"2019-07-23T06:21:46Z"
}
]
}
}
},
{
"Type":"Terrorism",
"TerrorismTask":{
"Status":"SUCCESS",
"ErrCode":0,
"Message":"",
"Input":{
"Definition":10
},
"Output":{
"Confidence":0,
"Suggestion":"pass",
"SegmentSet":[
]
}
}
},
{
"Type":"Political",
"PoliticalTask":{
"Status":"SUCCESS",
"ErrCode":0,
"Message":"",
"Input":{
"Definition":10
},
"Output":{
"Confidence":0,
"Suggestion":"pass",
"SegmentSet":[
]
}
}
}
],
"AiAnalysisResultSet":[
],
"AiRecognitionResultSet":[
]
},
"TasksPriority":0,
"SessionId":"",
"SessionContext":"",
"RequestId":"xxx-xxx-xxx"
}
```

As shown above, there are three types of results under `WorkflowTask.AiContentReviewResultSet` : `Porn` , `Terrorism` , and `Political` .

- For `Porn` , `Output.Suggestion` is `block` , which indicates a very high likelihood that the content is pornographic, and you are advised to block it. The confidence score is 98, and the label for the content is `sexy` .
- Three suspicious video segments are identified for `Porn` , whose start and end times are specified by `StartTimeOffset` and `EndTimeOffset` .
- According to the results for `Terrorism` and `Political` , no inappropriate content is detected in the video.

Error Codes

Last updated：2022-06-06 15:10:18

Media processing

Error Code	Description
InvalidInput	Invalid input parameter. Please check.
InvalidInput.InvalidTimeOffset	Invalid input parameter: the specified time point is invalid.
InvalidInput.DefinitionNotExist	Invalid input parameter: the specified template ID doesn't exist.
SourceFileError	Invalid source file (for example, video data is corrupted). Please check whether the source file is normal.
SourceFileError.NoVideoMedia	Invalid source file: there is no video image.
SourceFileError.NoVideoResolutio	Invalid source file: the resolution of the source file cannot be obtained.
InternalError	Internal service error. Please try again.