Media Processing Center

Best Practices

Issue 01

Date 2023-12-19

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Huawei Cloud Computing Technologies Co., Ltd.

Address:	Huawei Cloud Data Center Jiaoxinggong Road Qianzhong Avenue Gui'an New District Gui Zhou 550029 People's Republic of China
Website:	https://www.huaweicloud.com/intl/en-us/

Contents

1 Using MPC to Transcode Media Files in OBS

Scenarios

You may need to apply media files to different scenarios, such as the product official website and video website, or play the files on different terminals, such as the web client and mobile client. MPC provides transcoding to change the media encoding format, packaging format, resolution, and bitrate, so that media can be used in different scenarios, devices, and network environments.

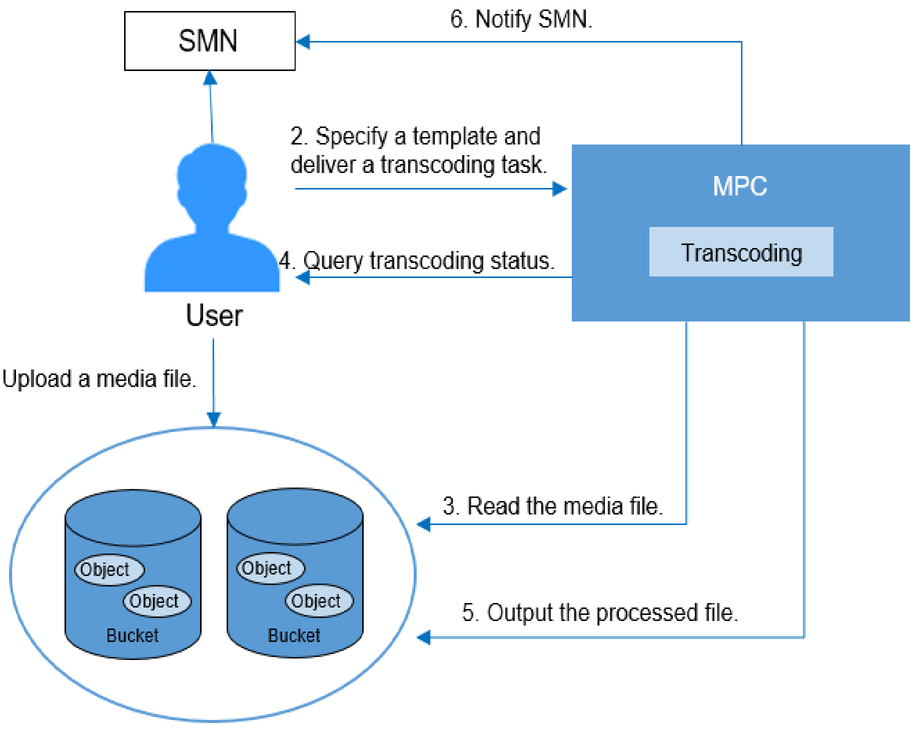
The transcoding function allows you to:

- Transcode source media files into formats such as MP4 for playback on a wide range of devices.
- Set the output bitrate based on the network bandwidth.
- Use H.265 codec and low bitrate HD to reduce the bitrate by about 20% without changing the resolution, thereby cutting media distribution costs.
- Enable HLS encryption during transcoding to prevent secondary distribution if a media file is stolen.
- Add watermarks such as logos to your video to protect copyright.
- Extract audio files through transcoding. This function is applicable to audio-only scenarios, such as radio stations and audio apps.
- Disable the original audio to output video-only files.

How It Works

For standard transcoding, you can use the default transcoding template to transcode media files in an OBS bucket and store the transcoded files in a specified OBS bucket. During transcoding, you can query the transcoding status. After the transcoding task is complete, a message is sent to you through SMN.

Figure 1-1 Transcoding on MPC



The process is as follows:

1. A user uploads the media file to be transcoded to OBS.
2. The user specifies an input/output transcoding template and delivers a transcoding task.
3. MPC obtains the media file specified by the user for processing.
4. The user periodically queries the transcoding status during transcoding.
5. After the transcoding task is complete, the transcoded media file is stored in the specified OBS directory.
6. The SMN service is used to notify users of transcoding status.
7. The user subscribes to a specified topic to obtain transcoding information.

Preparations

- The original media file has been uploaded to the OBS bucket, which is located in the region of MPC. If the file has not been uploaded, **upload the media file**.
- MPC has been authorized to access the buckets that store the input file and output file. If MPC has not been authorized, **authorize access to cloud resources**.
- If you want to send a message to notify the transcoding task execution status, **configure event notifications** first.
- If you want to use a custom template or template group for transcoding, **customize a transcoding template** or **customize a transcoding template group** first.

Creating a Video Transcoding Task

You can select a video transcoding template and create a video transcoding task to transcode video files stored in OBS buckets.

NOTE

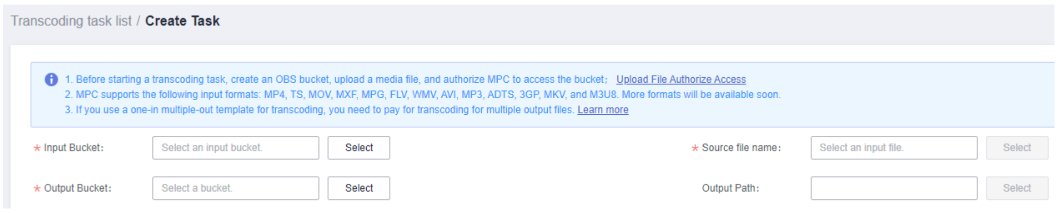
Video codecs supported are H.264, H.265, MPEG-2, MPEG-4, MJPEG, VP6/7/8/9, WMV1/2/3, and ProRes 422. If an input file is not in one of these formats, transcoding will fail.

Step 1 Log in to the MPC console.

Step 2 In the navigation pane, choose **Media Processing** > **Transcoding**.

Step 3 Click **Create Task**.

Step 4 Configure basic parameters, including the buckets and paths for storing an input file and output file.

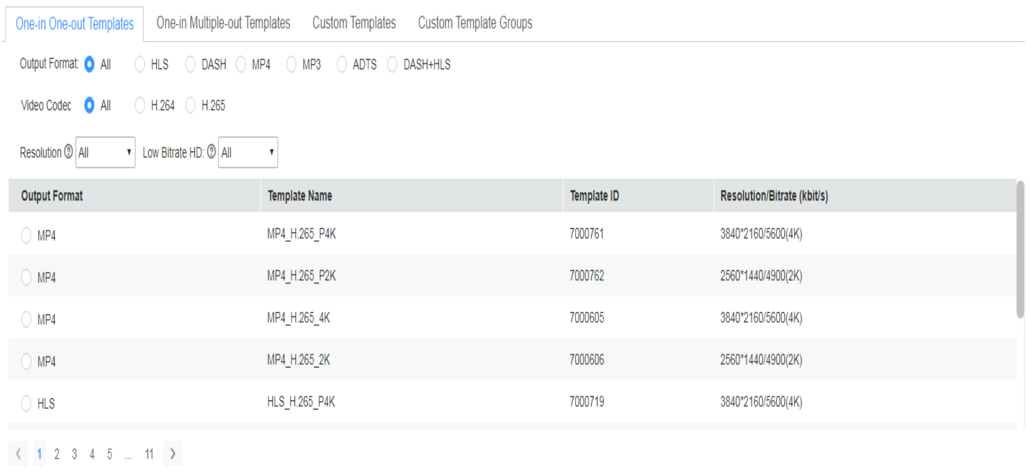


Transcoding task list / **Create Task**

1. Before starting a transcoding task, create an OBS bucket, upload a media file, and authorize MPC to access the bucket; Upload File Authorize Access
2. MPC supports the following input formats: MP4, TS, MOV, MXF, MPG, FLV, WMV, AVI, MP3, ADTS, 3GP, MKV, and M3U8. More formats will be available soon.
3. If you use a one-in multiple-out template for transcoding, you need to pay for transcoding for multiple output files. Learn more

* Input Bucket:	Select an input bucket.	Select	* Source file name:	Select an input file.	Select
* Output Bucket:	Select a bucket.	Select	Output Path:		Select

Step 5 Select a transcoding template as required.



Output Format: [✓ All] HLS DASH MP4 MP3 ADTS DASH+HLS

Video Codec [✓ All] H.264 H.265

Resolution: All Low Bitrate HD: All

Output Format	Template Name	Template ID	Resolution/Bitrate (kbit/s)
MP4	MP4_H.265_P4K	7000761	3840*2160/5600(4K)
MP4	MP4_H.265_P2K	7000762	2560*1440/4900(2K)
MP4	MP4_H.265_4K	7000605	3840*2160/5600(4K)
MP4	MP4_H.265_2K	7000606	2560*1440/4900(2K)
HLS	HLS_H.265_P4K	7000719	3840*2160/5600(4K)

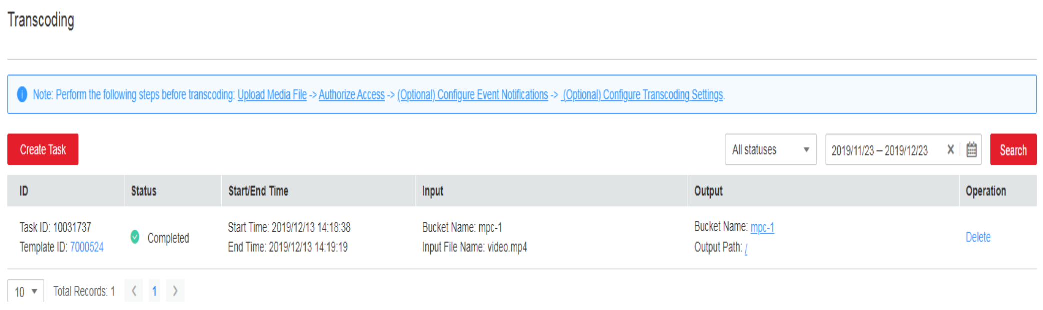
MPC provides a wealth of one-in one-out and one-in multiple-out system templates, which are configured with common parameters such as the definition, bitrate, and resolution. You are advised to use system templates. You can choose **Global Settings** > **System Templates** to view the parameters of a system template on the MPC console.

NOTICE

- Audio files cannot be transcoded using a video transcoding template.
- GIF files can be transcoded only to MP4 files.

Step 6 Click **OK**.

Step 7 View the transcoding task status in the task list. You can view details about transcoding tasks of the past 60 days.


Transcoding

Note: Perform the following steps before transcoding: Upload Media File -> Authorize Access -> (Optional) Configure Event Notifications -> (Optional) Configure Transcoding Settings.

Create Task | All statuses | 2019/11/23 – 2019/12/23 | Search

ID	Status	Start/End Time	Input	Output	Operation
Task ID: 10031737 Template ID: 7000524	Completed	Start Time: 2019/12/13 14:18:38 End Time: 2019/12/13 14:19:19	Bucket Name: mpc-1 Input File Name: video.mp4	Bucket Name: mpc-1 Output Path: /	Delete

10 Total Records: 1 < 1 >

- If transcoding succeeds, click **Output Path** in the **Output** column to switch to the OBS console, where you can view, download, and share the transcoded video file.
- If transcoding fails, view the failure cause in the **Output** column for troubleshooting.

----End

Creating an Audio Transcoding Task

You can select an audio transcoding template and create an audio transcoding task to transcode audio files stored in OBS buckets. The fee for audio transcoding is different from that for video transcoding. For details, see **Pricing Details**.

NOTE

Audio codecs supported are AAC, AC3, EAC3, HE-AAC, MP2, MP3, PCM (s161e, s16be, s241e, s24be, DVD), and WMA.

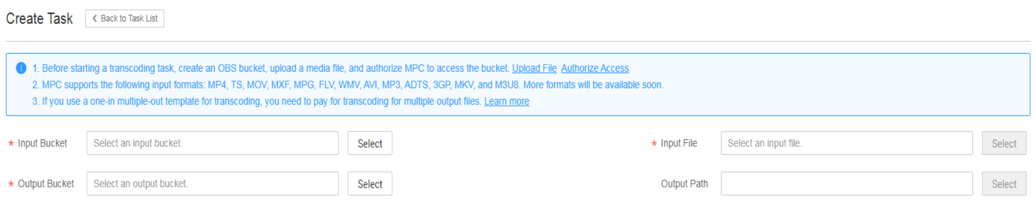
If an input file is not in one of these formats, transcoding will fail.

Step 1 Log in to the MPC console.

Step 2 In the navigation pane, choose **Media Processing** > **Transcoding**.

Step 3 Click **Create Task**.

Step 4 Configure basic parameters, including the buckets and paths for storing the input file and output file.

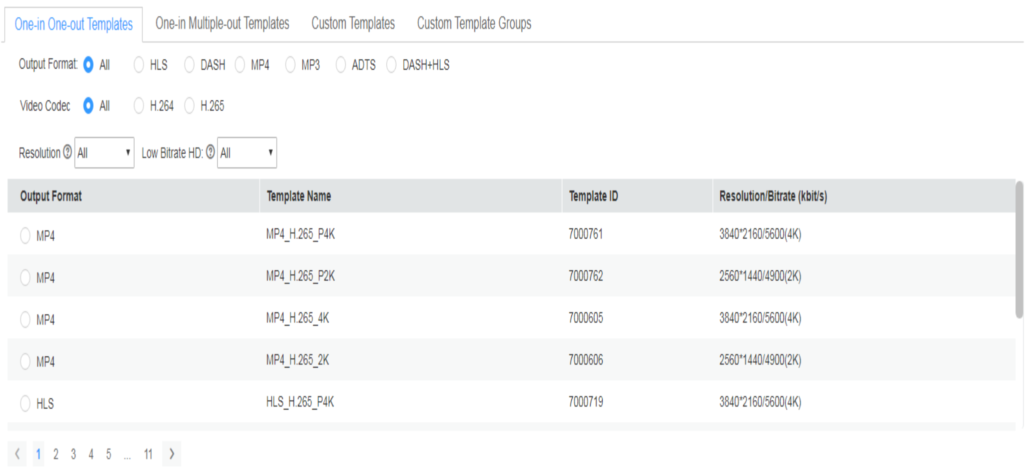

Create Task < Back to Task List

1. Before starting a transcoding task, create an OBS bucket, upload a media file, and authorize MPC to access the bucket. Upload File Authorize Access
2. MPC supports the following input formats: MP4, TS, MOV, MXF, MPG, FLV, WMV, AVI, MP3, ADTS, 3GP, MKV, and M3U8. More formats will be available soon.
3. If you use a one-in multiple-out template for transcoding, you need to pay for transcoding for multiple output files. Learn more

* Input Bucket	Select an input bucket.	Select	* Input File	Select an input file.	Select
* Output Bucket	Select an output bucket.	Select	Output Path		Select

Step 5 Select a transcoding template that best fits your needs.

- If you select **One-in One-out Templates**, select **MP3** or **ADTS** for **Output Format**.
- If you select **Custom Templates**, **create an audio transcoding template**.



One-in One-out Templates | One-in Multiple-out Templates | Custom Templates | Custom Template Groups

Output Format: [✓ All] HLS DASH MP4 MP3 ADTS DASH+HLS

Video Codec: [✓ All] H.264 H.265

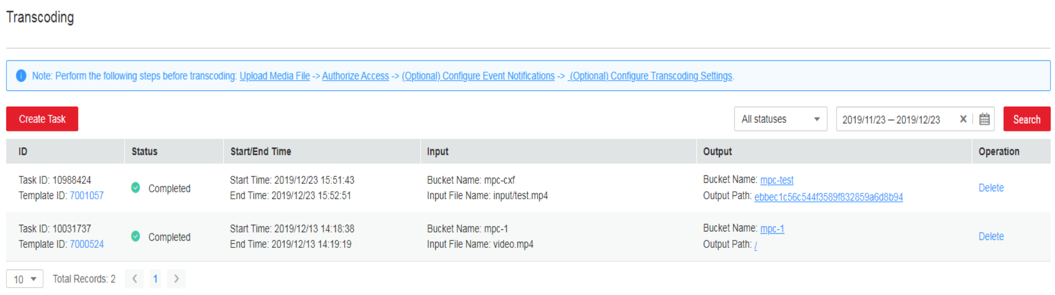
Resolution: All   Low Bitrate HD: All

Output Format	Template Name	Template ID	Resolution/Bitrate (kbit/s)
MP4	MP4_H.265_P4K	7000761	3840*2160/5600(4K)
MP4	MP4_H.265_P2K	7000762	2560*1440/4900(2K)
MP4	MP4_H.265_4K	7000605	3840*2160/5600(4K)
MP4	MP4_H.265_2K	7000606	2560*1440/4900(2K)
HLS	HLS_H.265_P4K	7000719	3840*2160/5600(4K)

< 1 2 3 4 5 ... 11 >

Step 6 Click **OK**.

Step 7 View the transcoding task status in the task list. You can view details about transcoding tasks of the past 60 days.



Transcoding

Note: Perform the following steps before transcoding: Upload Media File -> Authorize Access -> (Optional) Configure Event Notifications -> (Optional) Configure Transcoding Settings.

Create Task   All statuses   2019/11/23 – 2019/12/23   Search

ID	Status	Start/End Time	Input	Output	Operation
Task ID: 10988424 Template ID: 7001057	Completed	Start Time: 2019/12/23 15:51:43 End Time: 2019/12/23 15:52:51	Bucket Name: mpc-cxf Input File Name: input/test.mp4	Bucket Name: mpc-test Output Path: ebbec1c56c544f3589f832859a6d8b94	Delete
Task ID: 10031737 Template ID: 7000524	Completed	Start Time: 2019/12/13 14:18:38 End Time: 2019/12/13 14:19:19	Bucket Name: mpc-1 Input File Name: video.mp4	Bucket Name: mpc-1 Output Path: /	Delete

10   Total Records: 2   < 1 >

----End

2 H.264 and H.265 Low-bitrate HD Creates an Amazing Experience for Video Websites

Scenarios

Online video websites need to provide a good watching experience regardless of terminal types and bandwidth conditions. This is where MPC can come into play. Our codec algorithm optimizes the original video quality and reduces the video bitrate usage to ensure the video definition at a smaller video size. MPC lowers your storage and bandwidth costs, and improves metric performance such as video frame freezing, latency, and failure rate to enhance user experience.

When creating a transcoding task, you can enable the low-bitrate HD feature of MPC as needed. See the following for more details.

- **Configuration Method 1**
- **Configuration Method 2**

How It Works

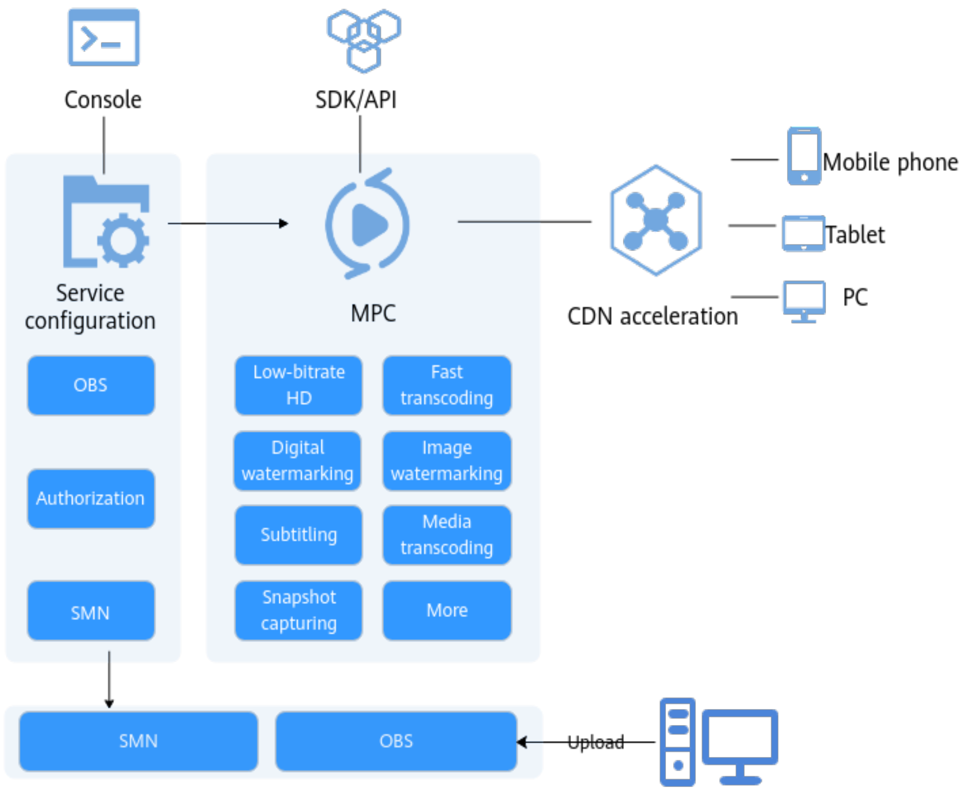


Figure 2-1 Working principle

Configuration Method 1

Step 1 Log in to the MPC console.

Step 2 In the navigation pane, choose **Media Processing** > **Transcode**.

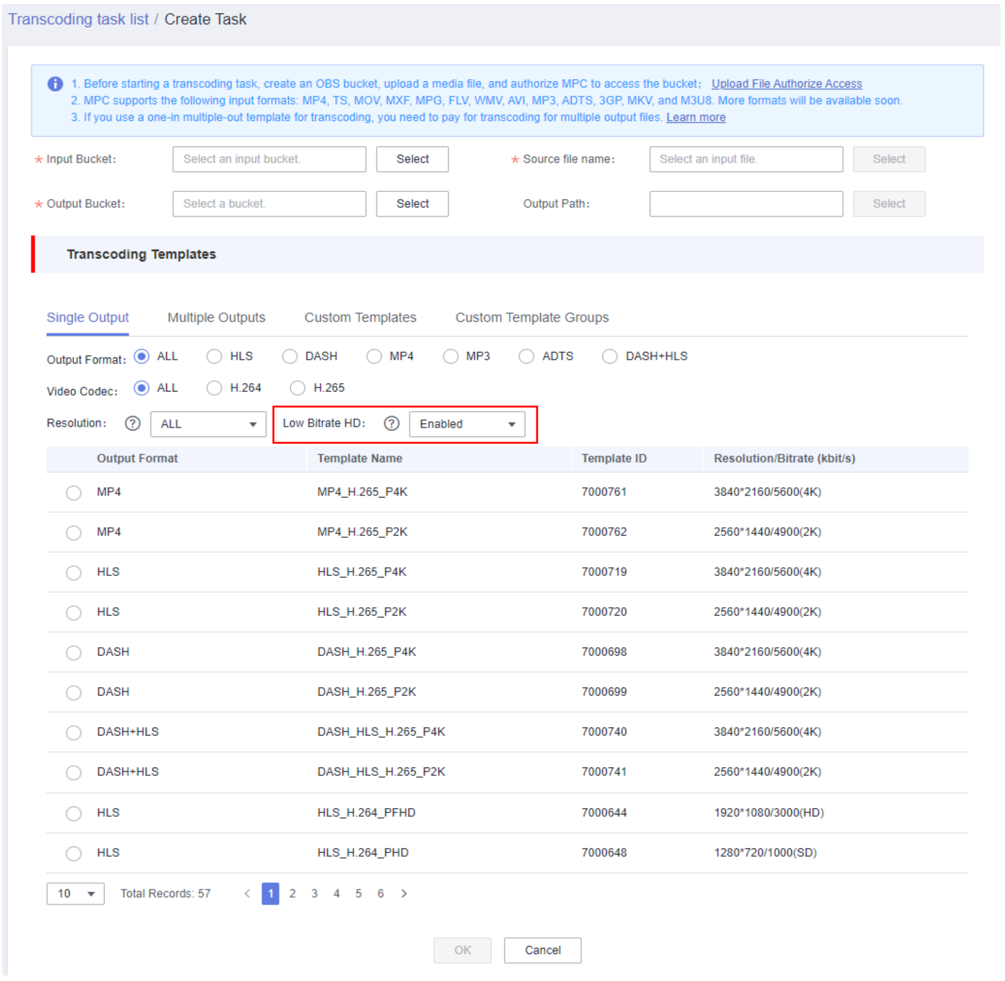
Step 3 Click **Create task**. The task creation page is displayed.

Step 4 Configure the **Low Bitrate HD** parameter of a transcoding template. Select **Enabled** from the drop-down list box, and then select the low-bitrate HD template, as shown in the following figure.

Notes:

- To create a low-bitrate HD template, set **PVC** in the request parameter **Common** to **true**. For details, see **Creating a Transcoding Template**.
- To create a low-bitrate HD template group, set **PVC** in the request parameter **Common** to **true**. For details, see **Creating a Transcoding Template Group**.

Figure 2-2 Low-bitrate HD templates



Transcoding task list / Create Task

1. Before starting a transcoding task, create an OBS bucket, upload a media file, and authorize MPC to access the bucket; Upload File Authorize Access
2. MPC supports the following input formats: MP4, TS, MOV, MXF, MPG, FLV, WMV, AVI, MP3, ADTS, 3GP, MKV, and M3U8. More formats will be available soon.
3. If you use a one-in multiple-out template for transcoding, you need to pay for transcoding for multiple output files. Learn more

* Input Bucket:	Select an input bucket.	* Source file name:	Select an input file.
* Output Bucket:	Select a bucket.	Output Path:	

Transcoding Templates

Single Output | Multiple Outputs | Custom Templates | Custom Template Groups

Output Format: [✓ ALL] HLS DASH MP4 MP3 ADTS DASH+HLS

Video Codec: [✓ ALL] H.264 H.265

Resolution: ALL

Low Bitrate HD: Enabled

Output Format	Template Name	Template ID	Resolution/Bitrate (kbit/s)
MP4	MP4_H.265_P4K	7000761	3840*2160/5600(4K)
MP4	MP4_H.265_P2K	7000762	2560*1440/4900(2K)
HLS	HLS_H.265_P4K	7000719	3840*2160/5600(4K)
HLS	HLS_H.265_P2K	7000720	2560*1440/4900(2K)
DASH	DASH_H.265_P4K	7000698	3840*2160/5600(4K)
DASH	DASH_H.265_P2K	7000699	2560*1440/4900(2K)
DASH+HLS	DASH_HLS_H.265_P4K	7000740	3840*2160/5600(4K)
DASH+HLS	DASH_HLS_H.265_P2K	7000741	2560*1440/4900(2K)
HLS	HLS_H.264_PFHD	7000644	1920*1080/3000(HD)
HLS	HLS_H.264_PHD	7000648	1280*720/1000(SD)

10 Total Records: 57 < 1 2 3 4 5 6 >

----End

Configuration Method 2

For the SDK that accesses the MPC, set **PVC** in the request parameter **Common** to **true** in the request body for creating a transcoding task, that is, **withPvc(true)**.

Sample core code:

```
    // Set the paths of the input and output videos.
    ObsObjInfo input = new ObsObjInfo().withBucket("mpc-east-2").withLocation("cn-
east-2").withObject("ok.mp4");
    ObsObjInfo output = new ObsObjInfo().withBucket("mpc-east-2").withLocation("cn-
east-2").withObject("output");

    AvParameters avParameters = new AvParameters();
    avParameters.setCommon(new Common().withPvc(true));
    ArrayList<AvParameters> avParametersArrayList = new ArrayList<>();
    avParametersArrayList.add(avParameters);

    CreateTranscodingTaskRequest request
        = new CreateTranscodingTaskRequest().withBody(new CreateTranscodingReq()
        .withInput(input)
        .withOutput(output)
        .withAvParameters(avParametersArrayList)
    );
    CreateTranscodingTaskResponse response = getMpcClient().createTranscodingTask(request);
    System.out.println("CreateTranscodingTaskResponse=" + response);
```

3 Snapshot Capturing Facilitates the Setup of Media Processing Platform for Your Video Website

Scenarios

Video websites have diverse requirements on video snapshots, including video thumbnails, drag and view, review, posters, and still images. MPC supports synchronous and asynchronous snapshot capturing, as well as snapshot capturing at a specified time point and at a fixed interval, allowing you to quickly set up a media processing platform for your video website. For example, during video playback, you can hover the pointer over the progress bar and drag it to go to the specified position on the preview image.

How It Works

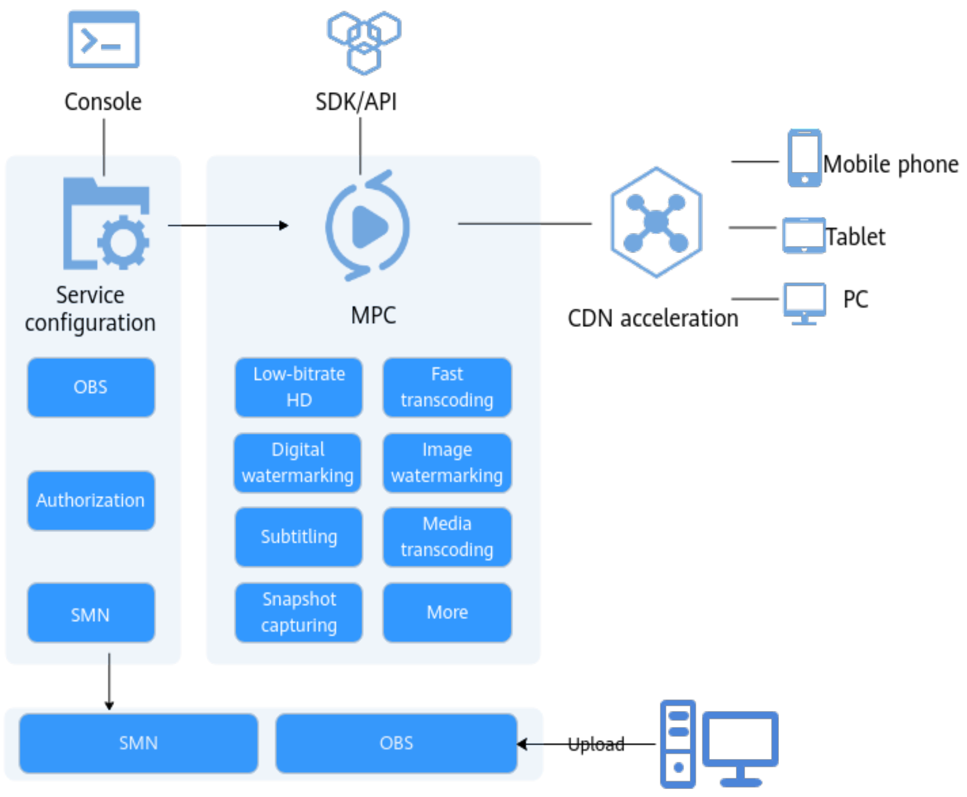
When setting up a media processing platform for your video website, you need to create a snapshot capturing task management service. This service manages operations such as creating and querying snapshot capturing tasks.

The video website snapshot capturing task management service calls the video snapshot capturing capability of MPC through SDKs/APIs. The snapshot capturing task management service obtains the source video from Object Storage Service (OBS). After a snapshot of the source video is captured as required, the snapshot file will be saved in a specified OBS path. The associated service of the video website can obtain the snapshot file information from the snapshot capturing task management service, and apply the snapshot to scenarios such as video thumbnails, drag and view during video playback, and review.

Table 3-1 Service functions

Service Name	Function
Video website snapshot capturing task management service	Manages video snapshot capturing tasks, including task creation and query.
MPC video snapshot capturing service	Pulls the source video from OBS, takes a snapshot of the source video as required, and saves the snapshot file in a specified OBS path.
OBS	Used by customers to upload and store media files

Figure 3-1 Working principle



Development Sequence Diagram

Figure 3-2 Creating a snapshot capturing task

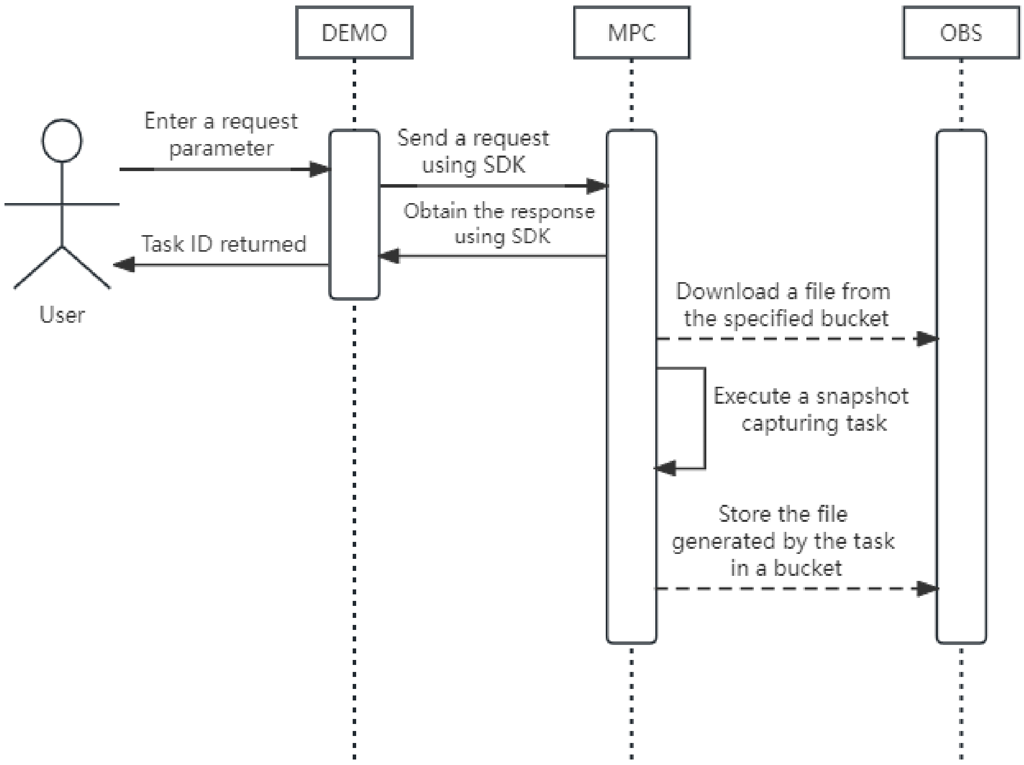

DEMO
MPC
OBS
User
Enter a request parameter
Send a request using SDK
Obtain the response using SDK
Task ID returned
Download a file from the specified bucket
Execute a snapshot capturing task
Store the file generated by the task in a bucket

Figure 3-3 Deleting a snapshot capturing task

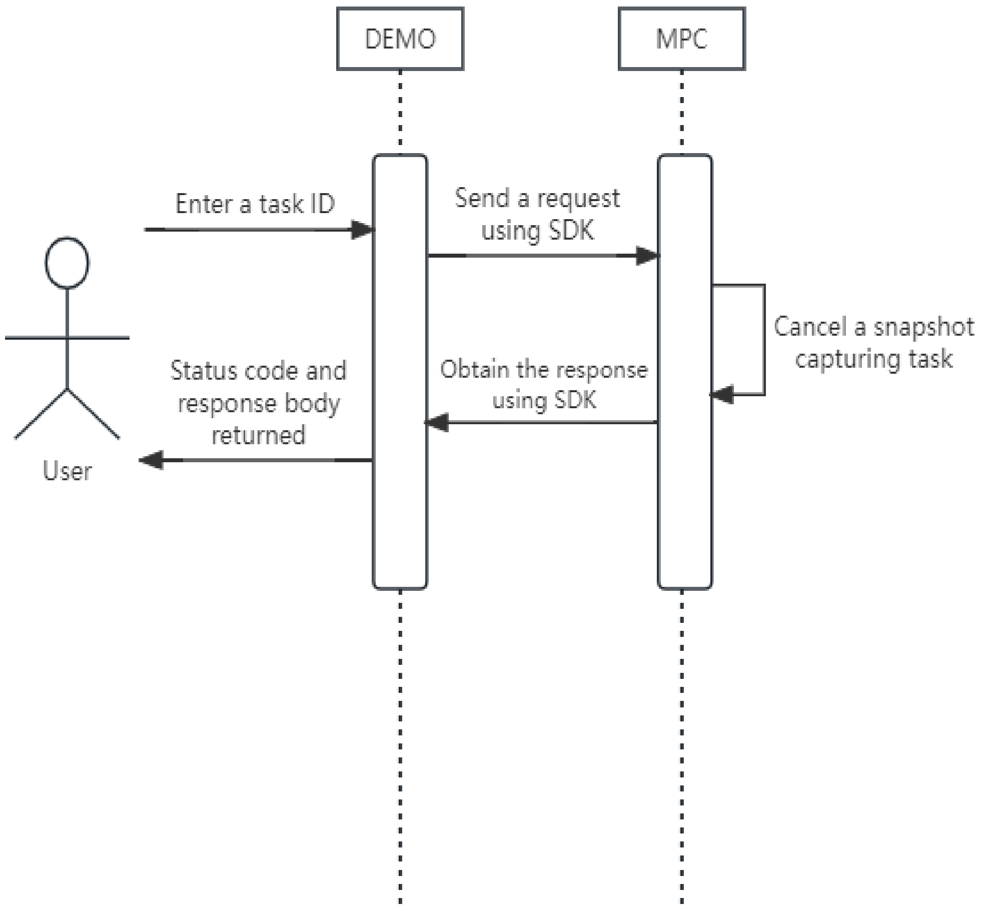
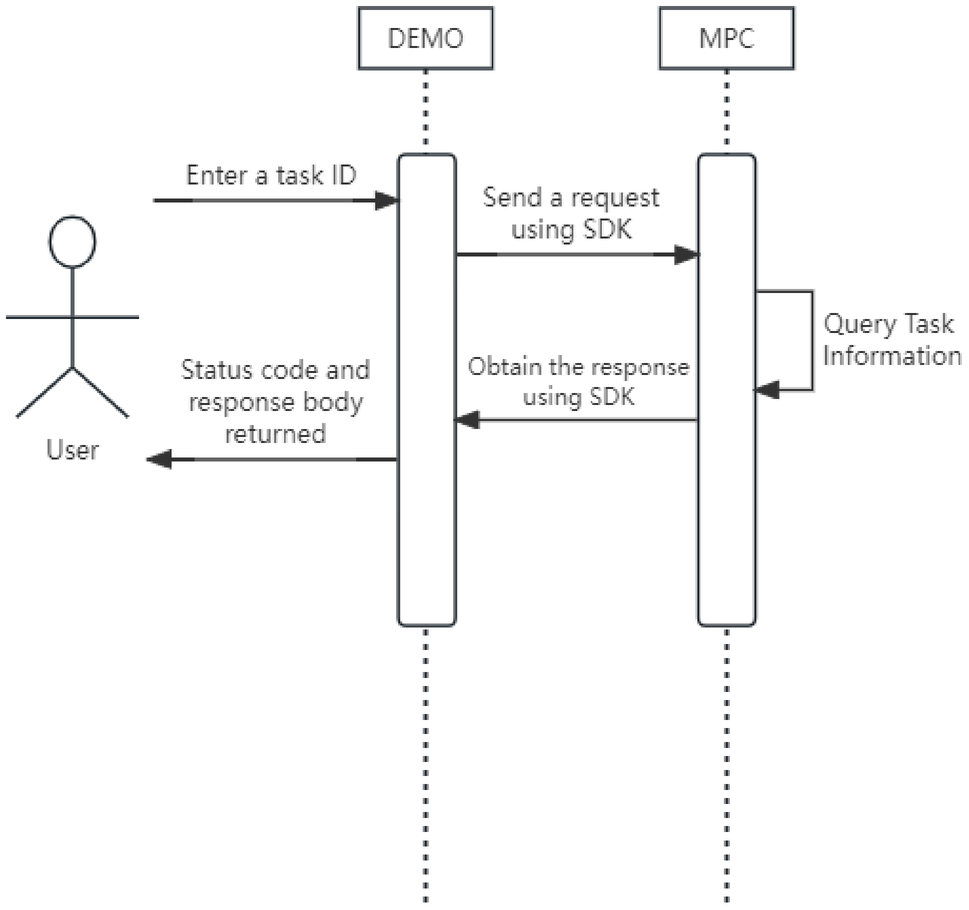

DEMO
MPC
Enter a task ID
Send a request using SDK
Cancel a snapshot capturing task
Obtain the response using SDK
Status code and response body returned
User

Figure 3-4 Querying snapshot capturing tasks



Procedure

Step 1 **Upload a video file** to an OBS bucket.

Step 2 Create a snapshot capturing task.

Sample code:

```
    // Set the input video path.
    ObsObjInfo input = new ObsObjInfo().withBucket("<example-bucket>").withLocation("<Region
ID>").withObject("<example-path/input.mp4>");
    ObsObjInfo output = new ObsObjInfo().withBucket("<example-bucket>").withLocation("<Region
ID>").withObject("<example-path/output>");

    // Create a snapshot capturing request.
    CreateThumbnailsTaskRequest request = new CreateThumbnailsTaskRequest();
    CreateThumbReq body = new CreateThumbReq();
    List<Integer> listThumbnailParaDots = new ArrayList<>();
    listThumbnailParaDots.add(50000);
    // Set the snapshot capturing type. Snapshots are captured at a specified time point.
    ThumbnailPara thumbnailParabody = new ThumbnailPara();
    // Set the sampling type. Snapshots are captured at a specified time point.
    thumbnailParabody.withType(ThumbnailPara.TypeEnum.fromValue("DOTS"))
        // Set the snapshot file name.
        .withOutputFilename("photo")
        // Set the interval for snapshot capturing.
        .withTime(10)
        // Set the start time when the sampling type is set to TIME. This parameter is used together with time.
        .withStartTime(100)
        // Set the capture duration (in second) when the sampling type is set to TIME. This parameter is used
together with time and start_time, indicating that the first snapshot is captured at the time specified by
start_time and a snapshot is captured at the interval specified by time until the duration specified by this
parameter elapses.
        .withDuration(1)
        // Array of time points when a snapshot is captured.
        .withDots(listThumbnailParaDots)
        // Set the snapshot file format.
        .withFormat(1)
        // Set the width of a snapshot.
        .withWidth(96)
        // Set the height of a snapshot.
        .withHeight(96);

    body.withThumbnailPara(thumbnailParabody);
    body.withOutput(output);
    body.withInput(input);
    request.withBody(body);

    // Send the snapshot capturing request.
    CreateThumbnailsTaskResponse response = initMpcClient().createThumbnailsTask(request);
    logger.info(response.toString());        return response.getTaskId();
```

Step 3 Query snapshot capturing tasks.

Sample code:

```
    ListThumbnailsTaskRequest request = new ListThumbnailsTaskRequest();
    List<String> listRequestTaskId = new ArrayList<>();
    listRequestTaskId.add(taskId);
    request.withTaskId(listRequestTaskId);
    ListThumbnailsTaskResponse response = initMpcClient().listThumbnailsTask(request);
    logger.info(response.toString());
```

Step 4 Check execution results.

The ID of the created snapshot capturing task is returned.

```
{"task_id": "1024"}
```

Query the statuses and results of snapshot capturing tasks.

```
{
  "task_array" : [
    {
    "task_id" : 2528,
    "status" : "SUCCEEDED",
    "create_time" : 20201118121333,
    "end_time" : 20201118121336,
    "input" : {
      "bucket" : "example-bucket",
      "location" : "region01",
      "object" : "example-input.ts"
    },
    "output" : {
      "bucket" : "example-bucket",
      "location" : "region01",
      "object" : "example-output/example-path"
    },
    "thumbnail_info" : [ {
      "pic_name" : "9.jpg"
    }, {
      "pic_name" : "5.jpg"
    } ]
  }
  ],
  "is_truncated" : 0,
  "total" : 1
}
```

----End

SDK Integration Example

For details about the video snapshot capturing feature and its sample code, see **Creating a Snapshot Capturing Task**.

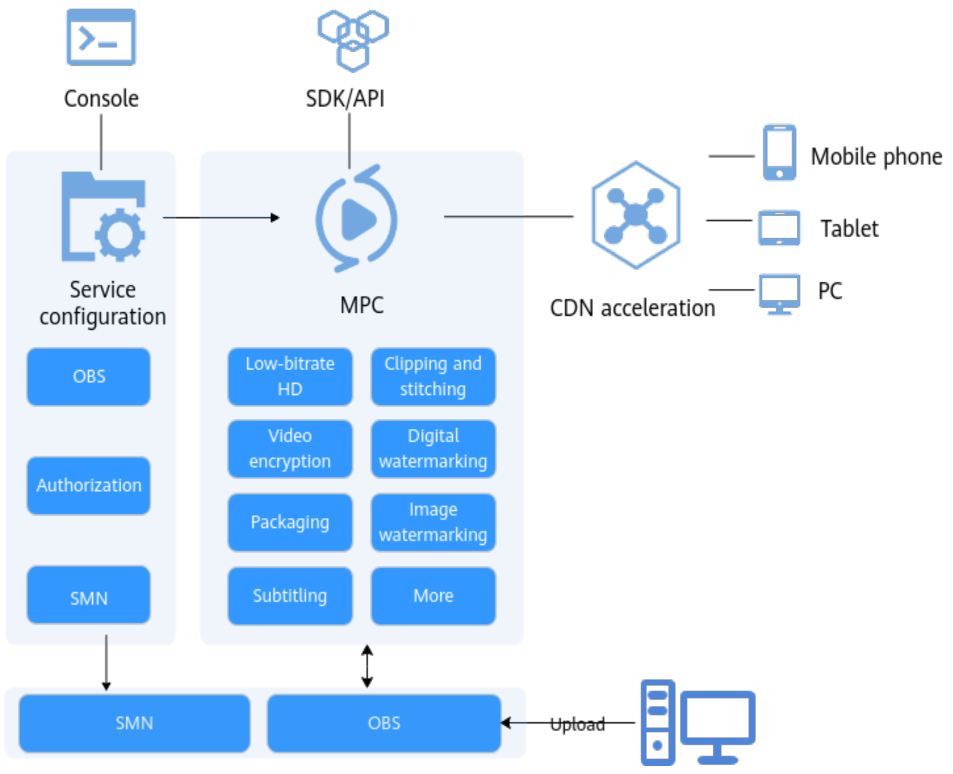
4 Video Packaging Enables the Playback of Online Education Videos on Multiple Terminal Types

Scenarios

MPC can convert mainstream video formats to MP4 or HLS for multi-terminal compatibility, so that online education videos can be played regardless of terminal types and network conditions. For example, users can watch videos of online education platforms on mobile apps.

How It Works

Figure 4-1 Working principle



Restrictions

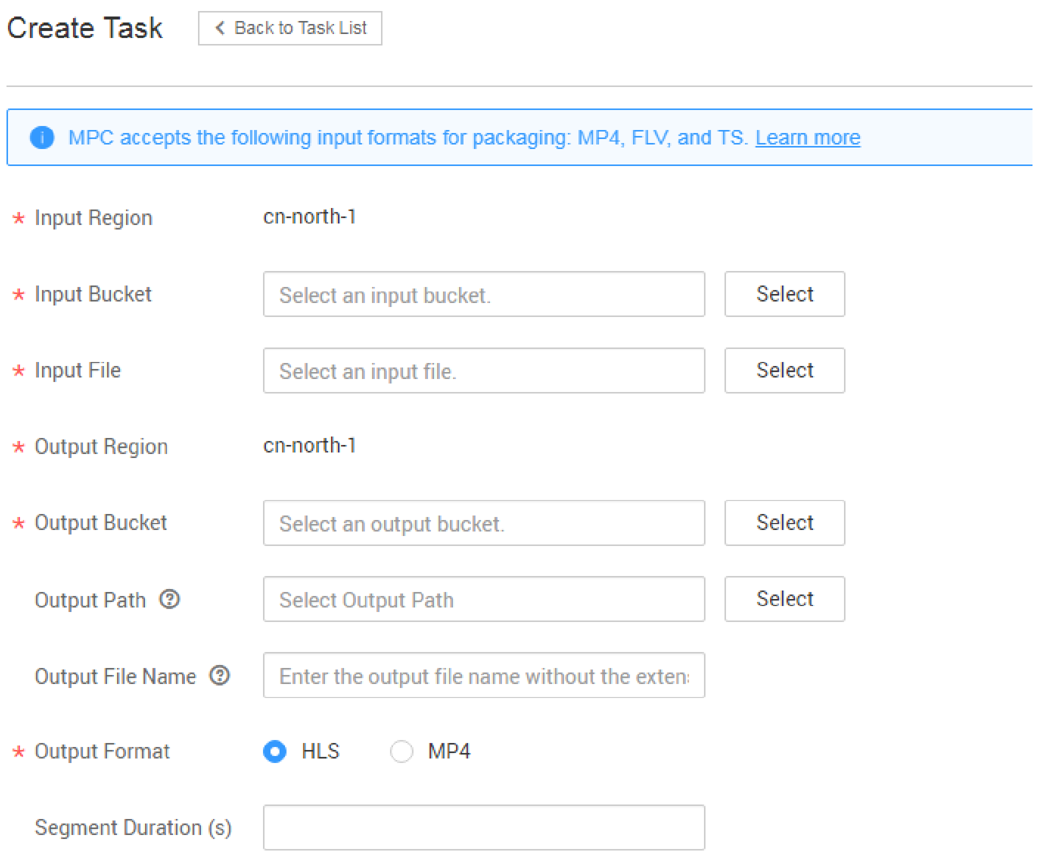
- Supported input formats: MP3, MP4, FLV, and TS
- Supported output formats: HLS and MP4

Procedure

Step 1 Log in to the MPC console.

Step 2 In the navigation pane, choose **Media Processing** > **Packaging**.

Step 3 Click **Create Task**.


Create Task
Back to Task List

MPC accepts the following input formats for packaging: MP4, FLV, and TS. Learn more

* Input Region	cn-north-1	
* Input Bucket	Select an input bucket.	Select
* Input File	Select an input file.	Select
* Output Region	cn-north-1	
* Output Bucket	Select an output bucket.	Select
Output Path	Select Output Path	Select
Output File Name	Enter the output file name without the exten:	
* Output Format	[✓ HLS] MP4	
Segment Duration (s)		

Step 4 Set task parameters by referring to **Table 4-1**.

Table 4-1 Task parameters

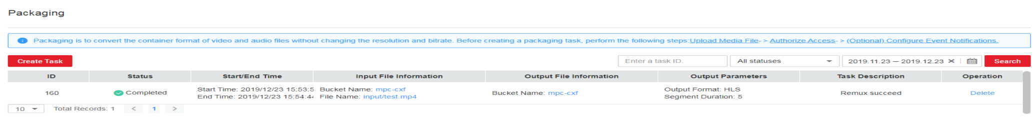
Parameter	Description
Input Region	Region where the OBS bucket for storing an input file resides
Input Bucket	OBS bucket where an input file is stored
Input File	Path for storing the input file

Parameter	Description
Output Region	Region where the OBS bucket for storing an output file resides
Output Bucket	OBS bucket where an output file is stored
Output Path	Path for storing the output file
Output File Name	Name of the packaged file
Output Format	Output format of the file. Currently, only the HLS and MP4 formats are supported.
Segment Duration (s)	HLS segment length. This parameter is only used when **Output Format** is **HLS**. The value ranges from 2 to 10. Default value: **5**

Step 5 Click **OK**.

Step 6 View the task status in the task list.

When the task status changes to **Completed**, you can obtain the packaged file from the output path.


Packaging

Packaging is to convert the container format of video and audio files without changing the resolution and bitrate. Before creating a packaging task, perform the following steps:Upload Media File- > Authorize Access- > (Optional) Configure Event Notifications.

ID	Status	Start/End Time	Input File Information	Output File Information	Output Parameters	Task Description	Operation
160	Completed	Start Time: 2019/12/23 15:53:5 End Time: 2019/12/23 15:54:4	Bucket Name: mpc-cxf File Name: input/test.mp4	Bucket Name: mpc-cxf	Output Format: HLS Segment Duration: 5	Remux succeed	Delete

----End

SDK Integration Example

For details about the packaging feature and its sample code, see **Creating a Packaging Task**.

SDK core code for education website developers to access MPC:

```
    ObsObjInfo input = new ObsObjInfo().withBucket("mpc-
east-2").withLocation("region01").withObject("ok.flv");
    ObsObjInfo output = new ObsObjInfo().withBucket("mpc-
east-2").withLocation("region01").withObject("output");
    // Create a packaging request.
    CreateRemuxTaskRequest req = new CreateRemuxTaskRequest()
            .withBody(new CreateRemuxTaskReq().withInput(input).withOutput(output)
                    // Configure packaging parameters.
                    .withOutputParam(new RemuxOutputParam()
                            // Set the packaging format.
                            .withFormat("HLS")
                            // Set the HLS segment interval.
                            .withSegmentDuration(5)));
    // Send the packaging request.
    CreateRemuxTaskResponse rsp = initMpcClient().createRemuxTask(req);
    System.out.println(rsp.toString())
```

5 Change History

Released On	Change Description
2023-11-30	This issue is the second official release. ● Added **H.264 and H.265 Low-bitrate HD Creates an Amazing Experience for Video Websites**. ● Added **Snapshot Capturing Facilitates the Setup of Media Processing Platform for Your Video Website**. ● Added **Video Packaging Enables the Playback of Online Education Videos on Multiple Terminal Types**.
2022-11-30	This issue is the first official release.