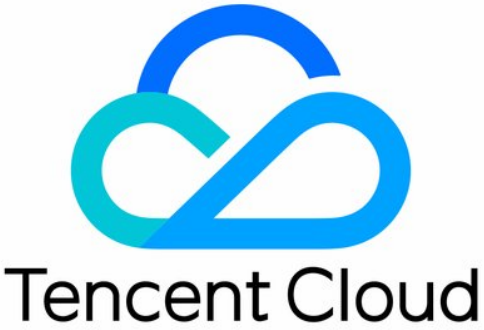
Optical Character Recognition
Client SDK Integration Guide



Copyright Notice

Trademark Notice

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Client SDK Integration Guide
Client SDK Integration Overview

Last updated: 2025-02-06 16:09:53

SDK Integration Overview

SDK Description

The Optical Character Recognition (OCR) client SDK provides a convenient, fast, and flexible way to integrate Tencent Cloud's OCR capabilities into your client, enabling you to use Tencent Cloud's quick and accurate OCR services.
Currently, the Tencent Cloud OCR client SDK supports the following capabilities:

- Identity card recognition
- Bank card recognition
- Business card recognition
- Vehicle VIN code recognition
- License plate recognition
- Vehicle license recognition
- Driving license recognition
- Hong Kong (China) identity card recognition
- Hong Kong (China), Macao (China), and Taiwan (China) permit recognition
- Hong Kong (China), Macao (China), and Taiwan (China) mainland travel permit recognition
- Passport recognition (Hong Kong (China), Macao (China), Taiwan (China), and foreign passports)

More OCR capabilities will be gradually added in the future, stay tuned!

Quick Experience

Tencent Cloud currently offers an OCR experience app for Android, Quick Experience in One Minute to experience the OCR client SDK capabilities.

SDK Integration

The OCR client SDK currently supports Android, iOS, and Harmony platforms. Go to the OCR console to download the SDK.

SDK Release Notes
IOS OCR Client SDK Release Notes

Last updated：2025-02-06 16:10:24

Fixed SDK Download URL

SDK Download

Update Content

v1.1.0.17 Update Content (July 17, 2024):
Fixed the issue of interface display exception when entering both sides of the mainland identity card.

v1.1.0.16 Update Content (July 1, 2024):
- Fix the issue with card parameter.
- Optimization of recognition range.
- Add transition animation control field isForbidAnima.

v1.1.0.15 Update Content (June 25, 2024):
Added Macao (China) identity document type (IDCardOCR_MACAU).

v1.1.0.14 Update Content (May 12, 2024):
Added configuration field minCropSize to support setting the minimum crop area.

v1.1.0.13 Update Content (March 12, 2024):
- Fix the issue where switching enhanced identity card recognition is ineffective.
- Optimize the memory release issue after recognition completion.
- Add image compression configuration field maxCompressLength.
- Fix the prompt text for both sides of the identity card.
- Adjust IDCardOCR_FRONT type to national emblem side and IDCardOCR_BACK type to portrait side.

v1.1.0.12 Update Content (March 10, 2024):
1. Fix the issue where the cropping page button did not support multilingual.
2. Fix the issue where the switch mode toast did not support multilingual.
3. Modify the pop-up to switch the cropping page to full-screen.

4. Modify the scan tips label constraints to increase the width.
5. Change the default value of validCount stable frame rate to 10 frames.

v1.1.0.11 Update Content (October 27, 2023):
Stability frame and blur threshold logic optimization.

v1.1.0.10 Update Content (July 10, 2023):
Add permission pop-up configuration.

v1.1.0.9 Update Content (June 21, 2023):
Fixed known issues.

v1.1.0.8 Update Content (June 12, 2023):
Added support for album-only mode switch.
Fixed known issues.

v1.1.0.7 Update Content (June 8, 2023):
Added configuration switch to disable cropping feature.

v1.1.0.6 Update Content (April 13, 2023):
Fixed known issues.

v1.1.0.5 Update Content (March 24, 2023):
Added configuration item: default camera zoom factor.

v1.1.0.4 Update Content (January 13, 2023):
- Add a field to control popup style.
- Optimize hidden button logic.
- Fixed known issues.

v1.1.0.3 Update Content (January 29, 2023):
- Fixed the alarm trigger issue in Xcode 14.
- Fixed known issues.

Android OCR Client SDK Update Log

Last updated：2025-02-06 16:10:36

Fixed SDK Download URL

SDK Download

Update Content

v2.0.0.19 (July 11, 2024)
Overseas version optimized for domestic identity card recognition

v2.0.0.18 (June 25, 2024)
Fixes known issues

v2.0.0.17 (May 27, 2024)
Adjust the preview size of the SDK in vertical screen

v2.0.0.16 (February 29, 2024)
Support minimum image capture size adjustment

v2.0.0.15 (February 26, 2024)
Resolve known issues

v2.0.0.12 (January 2, 2024)
Fixed the issue where some mobile phones could not select albums under targetSdkVersion 34

v2.0.0.11 (June 19, 2023)
Added showCustomPermissionDialog configuration item

v2.0.0.10 (June 19, 2023)
Added supportOnlyAlbumMode configuration item

Harmony NEXT Optical Character Recognition Client SDK Update Log

Last updated：2025-02-06 16:14:28

Fixed SDK Download URL

SDK Download

Update Content

v3.0.2.31 (September 11, 2024):
Corrected the authentication description and some incorrect names in the integration document

v3.0.0.14 (July 10, 2024):
First release

One – Minute Quick Experience

Last updated: 2025-02-06 16:14:37

Currently, Tencent Cloud offers an Optical Character Recognition experience app for Android. Scan the QR code to experience the OCR client SDK capabilities.

The Android OCR app provides the following capabilities for trial:

- Identity card recognition
- Bank card recognition
- Business card recognition
- Vehicle VIN code recognition
- License plate recognition
- Vehicle license recognition
- Driving license recognition

It also offers two recognition modes:

- Automatic recognition
- Photo Identification

Android OCR Experience App Page

Run Through the Demo In One Minute
Demo Quick Start (IOS)

Last updated：2025-02-06 16:15:01

This article mainly introduces how to quickly run the Tencent Cloud Optical Character Recognition (OCR) demo.

Environment Requirements

- Compatible with iOS 11.0 and later versions of the iOS system.
- Use Xcode 11.0 or later as the development tool, Xcode 11.5 is recommended.
- It is recommended to use an iPhone physical device for the experience.

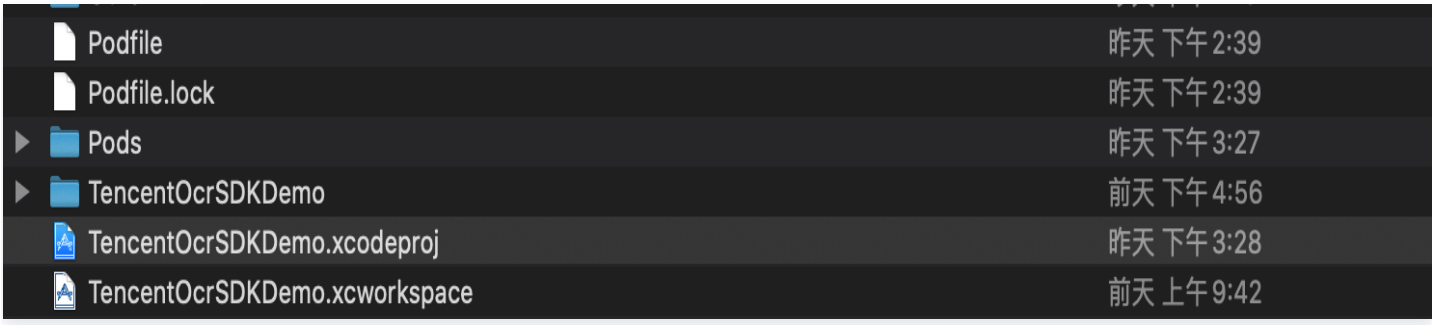
Prerequisites

- You have signed up for a Tencent Cloud account and completed real-name authentication.
- Log in to the OCR console to activate the service and obtain the API key in the account center.
- Go to the OCR client SDK Integration page to download the corresponding SDK and download the SDK demo files to your local machine.

Operation Steps

Step 1. Import the Project

Enter the downloaded demo folder and double-click to open the project engineering with Xcode.



Step 2. If Using Real Device Testing, Configure Your Apple Developer Account

TARGETS > Signing & Capabilities > Signing settings

Step 3. Modify Configuration Information

Modify the SecretId and SecretKey values in the project and enter your API key from the Tencent Cloud Console.

```
//Formal
static NSString *SECRET_ID = @"A**********************";
static NSString *SECRET_KEY = @"e**********************";
```

Note

The method of directly configuring secretId and secretKey in the client mentioned in this article is easily decompiled and reverse cracked. Once your keys are leaked, attackers can use your secretId and secretKey information to make OCR identify requests, causing you losses. Therefore, this method **is only suitable for running demos locally and debugging features.**

We recommend configuring secretId and secretKey on the server side, while the client uses temporary keys for OCR identification. For specific steps, refer to the Temporary Key Exchange Process.

Step 4. Compile and Run

After completing the configuration, select your running device (emulator & physical device). (Tips: Only a physical device can experience the camera photo recognition feature.)

Demo Quick Start (Android)

Last updated: 2025-02-06 16:15:13

This article mainly introduces how to quickly run the Tencent Cloud Optical Character Recognition (OCR) demo.

Environment Requirements

- Android 4.4 (SDK API level 19) or later is required (Android 5.0 (SDK API level 21) or later is recommended).
- Android Studio 4.0 or above
- The app requires devices with Android 4.4 or above

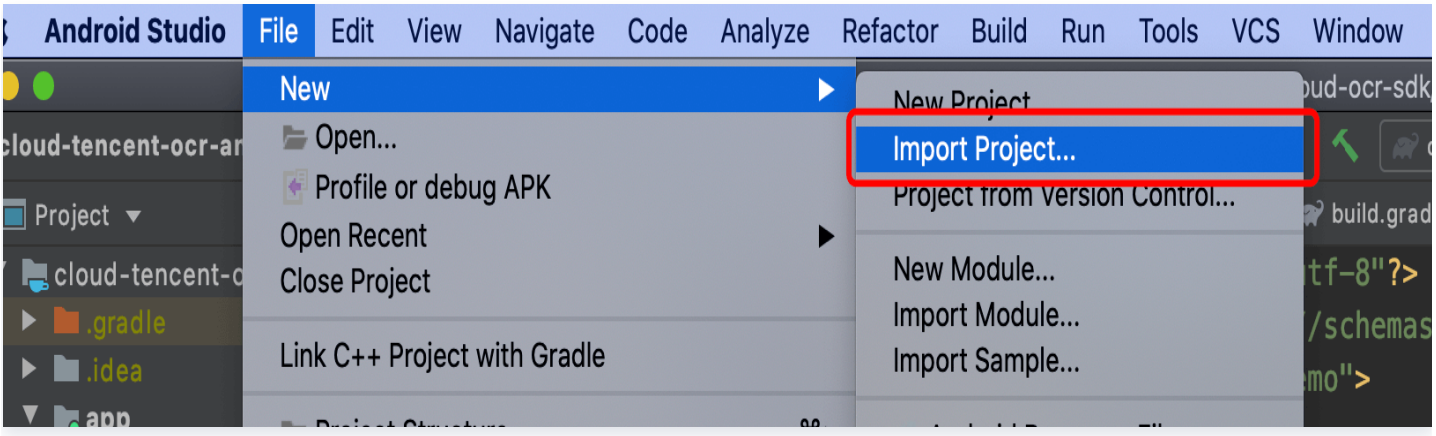
Prerequisites

- You have signed up for a Tencent Cloud account and completed real-name authentication.
- Log in to the OCR console to activate the service and obtain the API key in the account center.
- Go to the OCR client SDK Integration page to download the corresponding SDK and download the SDK demo files to your local machine.

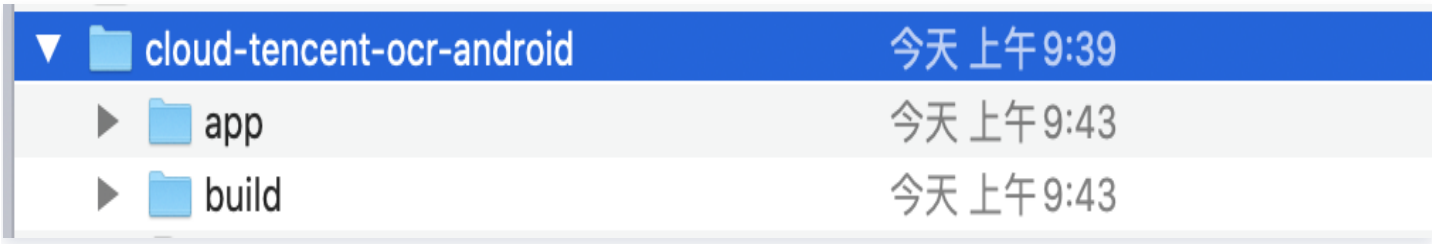
Operation Steps

Step 1. import the project

Open Android Studio and select the import project option.



Then select the demo project and import it.



Step 2. modify the configuration information

Open the file SecretPamera class and update the secretId and secretKey obtained in the background with your information.

```
/**
 * Key Configuration Information
 */
public class SecretPamera {
    public final static String secretId = "your secretId";
    public final static String secretKey = "your secretKey";
}
```

Note

The method mentioned in this article of directly configuring secretId and secretKey into the client is easily decompiled and reverse cracked. Once your keys are leaked, attackers can use your secretId and secretKey information to make OCR identify requests, causing you loss. Therefore, this method **is only suitable for locally running demos and feature debugging**.

> We recommend configuring secretId and secretKey on the server side, while the client uses temporary keys for OCR identification. For specific steps, refer to the Temporary Key Exchange Process.

Step 3. compile and run

After completing the configuration, you can click the run button in Android Studio to experience the demo.

Running Demo (Harmony)

Last updated: 2025-02-06 16:15:21

This article mainly introduces how to quickly run the Tencent Cloud Optical Character Recognition (OCR) demo.

Environment Requirements

- Minimum compatible phone system API level 11.
- DevEco version 5.0.3 and above (it is recommended to use the latest development tool).

Prerequisites

- You have registered a Tencent Cloud account and completed real-name verification.
- Log in to the OCR console to activate the service and obtain the API key in the account center.
- Go to the OCR client SDK Integration page to download the corresponding SDK and download the SDK demo file locally.

Operation Steps

Step 1: Open the Demo Project

Use the DevEco development tool to import the demo source code from the SDK delivery package.

Step 2: Modify the Configuration Information

Open the file OcrCardSelectPage.ets and update the secretId and secretKey obtained in the background with your information.

```
let config: OcrSDKConfig = new OcrSDKConfigBuilder()
  .setSecretID(secretId)
  .setSecretKey(secretKey)
```

Note

The method mentioned in this article of directly configuring secretId and secretKey into the client is easily decompiled and reverse cracked. Once your keys are leaked, attackers can use your secretId and secretKey information to make OCR identify requests, causing you loss. Therefore, this method **is only suitable for locally running demos and feature debugging**.

We recommend configuring secretId and secretKey on the server side, while the client uses temporary keys for OCR identification. For specific steps, refer to the Temporary Key Exchange Process.

Step 3: Compile and Run the Demo

After completing the configuration, you can click the run button in DevEco to experience the demo.

Integrate the SDK In One Minute Quick Integration (IOS)

Last updated：2025-02-06 16:15:46

Development Preparation

1. Sign up for a Tencent Cloud account and click to enter the Optical Character Recognition Console to activate the service.
2. Obtain the API key in the account center.
3. Go to the Optical Character Recognition client SDK Integration page to download the corresponding SDK.

IOS OCR SDK Access Process

Introduction To OCR SDK For IOS

The SDK includes the following framework libraries and resource files:

- **OcrSDKKit.framework** – OCR external API, page settings, and network request library
- **YTImageRefiner.framework** – Image parsing
- **tiny_opencv2.framework** – opencv library
- **tnn.framework** – underlying deep learning library
- **OcrSDK.bundle** – resource file
- **Photos.framework** – photo library

Environment Dependency

- The current iOS OCR Recognition SDK version is applicable to iOS 11.0 or later versions
- Use development tool xcode11 or later version for integration development

Access Steps

1. Add the OCR framework, system framework libraries, and bundle file to the project.

```
├── OcrSDKKit.framework
├── YTImageRefiner.framework
├── tiny_opencv2.framework
└── tnn.framework
// system database
├── Accelerate.framework
├── Photos.framework
├── PhotosUI.framework
└── CoreML.framework
```

```
// resource file
└── OcrSDK.bundle
```

2. Add compilation options.
   - Set **the ViewController calling the SDK** to **Objective-C++ Source** or change the suffix to **.mm** (Objective-C++ syntax is used internally in the SDK)
   - Add **-ObjC** in **Other Linker Flags**
3. Permission settings
   OCR SDK requires permissions for mobile network, camera, and access to album. Please add the corresponding permission statements.

```
<key>Privacy - Camera Usage Description</key>
<string>OCR requires enabling your camera permission for
recognition</string>
<key>Privacy - Photo Library Usage Description</key>
<string>OCR requires enabling your album permission to browse your
photos</string>
<key>Prevent limited photos access alert</key>
<Boolean>YES</Boolean>
```

SDK API Description

SDK initialization

Customer initializes the OCR SDK

```
#import <OcrSDKKit/OcrSDKKit.h>
#import <OcrSDKKit/OcrSDKConfig.h>

static NSString* const SECRET_ID     = @""; // SECRET_ID information
static NSString* const SECRET_KEY    = @""; // SECRET_KEY information
/*
 * OCR configuration class:
 * ocrModeType: detection type OCR_DETECT_MANUAL manual shoot;
OCR_DETECT_AUTO_MANUAL automatic card recognition
 */
OcrSDKConfig *ocrSDKConfig = [[OcrSDKConfig alloc] init];
ocrSDKConfig.ocrModeType = _ocrModel;
/// SDKKIt loads OCR configuration information
/// @param secretId  Secret id
/// @param secretKey Secret key
/// @param ocrConfig OCR configuration class
[[OcrSDKKit sharedInstance] loadSDKConfigWithSecretId:nil
withSecretKey:nil withConfig:ocrSdkConfig];
```

Entering the ocr homepage

```
/*!
OCR UI configuration class:
*/
CustomConfigUI *customConfigUI = [[CustomConfigUI alloc] init];
customConfigUI.remindConfirmColor = [UIColor blueColor];
/// Launch the SDK module, run the feature recognition module with a UI
interface
/// @param OcrType recognition mode
/// @param customConfigUI UI configuration object
/// @param onProcessSucceed success callback block
/// @param onProcessFailed failure callback block
[[OcrSDKKit sharedInstance] startProcessOcr:IDCardOCR_BACK
withSDKUIConfig:customConfigUI withProcessSucceedBlock:^(id  _Nonnull
resultInfo, UIImage *resultImage, id  _Nonnull reserved) {
    ///resultInfo recognition successful information (json)
  ///resultImage image captured after recognition successful
} withProcessFailedBlock:^(NSError * _Nonnull error, id  _Nullable
reserved) {
  ///error error information
  ///reserved usually returns requestid to locate errors
```

```
}];
```

Updating the temporary key

The OCR SDK supports using temporary key APIs. The main benefits of using temporary keys are: first, separating fixed keys from the terminal can enhance security; second, since the exchange of temporary keys is entirely under your control, you can control the API access permissions of end users based on custom rules. Therefore, it is recommended to use temporary keys. For details, refer to the documentation (Temporary Key Documentation and Process Link)

```
/// @param tmpSecretId temporary SecretId
/// @param tmpSecretKey temporary key information
/// @param token temporary exchange token
[[OcrSDKKit sharedInstance] updateFederationToken:tmpSecretId
withTempSecretKey:tmpSecretKey withToken:token];
```

SDK resource release

```
/// clear SDK resources
[OcrSDKKit clearInstance];
```

Currently, the OCR SDK supports seven types of recognition modes as shown in the table below.

OcrType Type	Representing Meaning
OcrType.IDCardOCR_FRONT	ID card portrait side recognition mode
OcrType.IDCardOCR_BACK	ID card national emblem side recognition mode
OcrType.BankCardOCR	Bank card front recognition mode
OcrType.BusinessCardOCR	Business card front recognition mode
OcrType.MLIdCardOCR	Malaysian ID card recognition mode
OcrType.LicensePlateOCR	Vehicle license plate recognition mode

OcrType.VinOCR	Vehicle VIN code recognition mode
OcrType.VehicleLicenseOCR_FRONT	Driving license home page recognition mode
OcrType.VehicleLicenseOCR_BACK	Driving license secondary page recognition mode
OcrType.DriverLicenseOCR_FRONT	Driving license home page recognition mode
OcrType.DriverLicenseOCR_BACK	Driving license secondary page recognition mode
OcrType.VinOCR2	Vehicle VIN code recognition mode (high-precision version)
OcrType.HKIDCardoCR_03	Hong Kong (China) ID card version 03 recognition mode
OcrType.IDCardOCR_HK18	Hong Kong (China) ID card version 18 recognition mode
OcrType.PermitOCR	Hong Kong (China), Macao (China), and Taiwan (China) travel permit recognition mode
OcrType.MLIDPassportocR	International passport recognition mode
OcrType.HmtResidentPermitOCR	Residence permit for Hong Kong (China), Macao (China), and Taiwan (China)

Common Mistakes

1. When prompted **requsetConfigDict is nil**, check if [OcrSDKKit cleanInstance] was executed when entering the SDK, which cleared the keys and configuration settings.
2. The SDK page relies on UIWindow, so you need to add **@property (nonatomic, strong) UIWindow * window;** in AppDelegate.h.
3. When a black screen appears upon entering the SDK, add **Other Linker Flags** and **-ObjC**. The log prints **Application tried to push a nil view controller on target....**, indicating that self.storyboard is nil. You can refer to the demo, manually load the xib page in the ViewController calling the SDK page, and then call the SDK to enter the recognition page.

4. Building for iOS Simulator, but the linked and embedded framework 'OcrSDKKit.framework' was built for iOS + iOS Simulator.
   The solution is: Build Settings --> Build Options --> Validate Workspace set to Yes.

Quick Integration (Android)

Last updated：2025-02-06 16:15:57

Development Preparation

1. Sign up for a Tencent Cloud account and click to enter the Optical Character Recognition Console to activate the service.
2. Obtain the API key in the account center.
3. Go to the Optical Character Recognition client SDK Integration page to download the corresponding SDK.

Android OCR SDK Access Process

Introduction To OCR SDK For Android

The file provided by the SDK is OCR_Android_SDK_V1.0.9, which encapsulates OCR recognition terminal capabilities. It currently includes identity card recognition, bank card recognition, business card recognition, vehicle Vin code recognition, license plate recognition, driving license recognition, and driving license recognition.

Environment Dependency

The current OCR recognition SDK for Android is supported by API 19 (Android 4.4) or later.

Access Steps

1. Add **OcrSDK-public-v2.0.0.1-release.aar**, **OcrSDK-common-model-v1.0.0-release.aar**, and the public libraries **tencent-ai-sdk-aicamera-1.0.18-release.aar**, **tencent-ai-sdk-common-1.1.27-release.aar** to the libs directory in your project directory.
2. Configure the **build.gradle** in your project as follows:

```
dependencies {
  // Depend on Tencent Cloud's OcrSDK aar
  implementation files('libs/OcrSDK-common-model-v1.0.0-release.aar')
  implementation files('libs/OcrSDK-public-v2.0.0.1-release.aar')
  implementation files('libs/tencent-ai-sdk-aicamera-1.0.18-
release.aar')
  implementation files('libs/tencent-ai-sdk-common-1.1.28-
release.aar')
  // Dependency required for OCR SDK return entity object
  implementation 'com.google.code.gson:gson:2.8.5'
}
```

3. Necessary permission declarations are also required in the AndroidManifest.xml file.

```
<!--Usage permission for camera-->
<uses-feature android:name="android.hardware.camera" />
<uses-permission
  android:name="android.permission.CAMERA"
  android:required="true" />
<!--Usage permission for Cloud File Storage [optional]-->
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<!--Network access permission-->
<uses-permission android:name="android.permission.INTERNET" />
```

For users who need to be compatible with Android 6.0 and above, the above permissions need to be declared in the AndroidManifest.xml file and dynamically requested using code.

SDK APIs Use Instructions

SDK initialization

Initialization is required before using the OCR SDK, and you can set the default values according to your needs.

```
// Startup parameter configuration
OcrModeType modeType = OcrModeType.OCR_DETECT_AUTO_MANUAL; // Set the
default business identification mode to automatic + manual steps mode
OcrType ocrType = OcrType.BankCardOCR; // Set the default business
identification to bank card
```

```
OcrSDKConfig configBuilder =
OcrSDKConfig.newBuilder(SecretPamera.secretId, SecretPamera.secretKey,
null)
    .setOcrType(ocrType)
    .setModeType(modeType)
    .build();
// Initialize the SDK
OcrSDKKit.getInstance().initWithConfig(this.getApplicationContext(),
configBuilder);
```

The business modes currently supported by the OCR SDK are:

<OcrModeType Type>	Representation Of Meaning
OCR_DETECT_MANUAL	Manual shooting mode
OCR_DETECT_AUTO_MANUAL	Automatic Identification mode (Note: After 20s, prompt to switch to manual shooting)

Updating the temporary key

The OCR SDK supports using temporary key APIs. The main benefits of using temporary keys are: first, separating fixed keys from the terminal can enhance security; second, since the exchange of temporary keys is entirely under your control, you can control the API access permissions of end users based on custom rules. Therefore, it is recommended to use temporary keys. For details, refer to the documentation (Temporary Key Documentation and Process Link)

```
OcrSDKKit.getInstance().updateFederationToken(tmpSecretId, tmpSecretKey,
token);
```

OCR recognition (returns json string)

When you need to use the OCR recognition feature, you can directly call the recognition API for OCR business recognition.

```
// Start OCR recognition, recognition type is ID card front
OcrSDKKit.getInstance().startProcessOcr(MainActivity.this,
OcrType.IDCardOCR_FRONT, customConfigUi, new ISDKKitResultListener() {
    @Override
    public void onProcessSucceed(String response, String srcBase64Image,
String requestId) {
```

```
        popTip(response, "Succeed"); // Display OCR recognition result
    }

    @Override
    public void onProcessFailed(String errorCode, String message, String
requestId) {
        popTip(message, errorCode); // Display OCR recognition error
information
    }
});
```

OCR recognition (returns object entity)

When you need to use the OCR recognition feature and directly obtain the entity object instead of JsonString, you can use this method.

```
OcrSDKKit.getInstance().startProcessOcrResultEntity(OcrTypeIdCardActivit
y.this,
        OcrType.IDCardOCR_FRONT, null, IdCardOcrResult.class,
        new ISdkOcrEntityResultListener<IdCardOcrResult>() {
            @Override
            public void onProcessSucceed(IdCardOcrResult
idCardOcrResult, String base64Str) {
                Log.e(TAG, "IdCardOcrResult:" +
idCardOcrResult.toString()); // OCR recognition succeeded
IdCardOcrResult
            }

            @Override
            public void onProcessFailed(String errorCode, String
message, String requestId) {
                Log.e(TAG, "errorCode:" + errorCode + " message:" +
message); // OCR recognition failed IdCardOcrResult
            }
        });
```

Currently, the OCR SDK supports several types of recognition modes as shown in the table below, along with the corresponding entity class return results.

OcrType Type	Representing Meaning	Corresponding Result Entity Class

OcrType.IDCardOCR_FRONT	ID card portrait side recognition mode	IdCardOcrResult
OcrType.IDCardOCR_BACK	ID card national emblem side recognition mode	IdCardOcrResult
OcrType.BankCardOCR	Bank card front recognition mode	BankCardOcrResult
OcrType.BusinessCardOCR	Business card front recognition mode	BusinessCardOcrResult
OcrType.VinOCR	Vehicle VIN recognition mode	VinOcrResult
OcrType.LicensePlateOCR	Vehicle license plate recognition mode	CarLicensePlateResult
OcrType.DriverLicenseOCR_FRONT	Driving license home page recognition mode	DriverLicenseCardResult
OcrType.DriverLicenseOCR_BACK	Driving license secondary page recognition mode	DriverLicenseCardResult
OcrType.VehicleLicenseOCR_FRONT	Driving license home page recognition mode	VehicleLicenseCardResult
OcrType.VehicleLicenseOCR_BACK	Driving license secondary page recognition mode	VehicleLicenseCardResult
OcrType.GENERAL_VIN	General vehicle vin code recognition mode (mainly recommended for photo mode)	VinOcrResult
OcrType.IDCardOCR_HK03	Hong Kong (China) identity card version 03 recognition mode	HKIDCardOcrResult
OcrType.IDCardOCR_HK18	Hong Kong (China) identity card version 18 recognition mode	HKIDCardOcrResult
OcrType.Exit_Entry_HK_Macao_Card	Hong Kong (China), Macao (China), and Taiwan (China) travel permit recognition mode	PermitOcrResult
OcrType.MLID_PASSPORT	International passport recognition mode	MLIDPassportOcrResult
OcrType.HMT_RESIDENT_PERMIT_OCR	Residence permit for Hong Kong (China), Macao (China), and	HmtResidentPermitOcrResult

	Taiwan (China)	

Get the sdk version number

OCR SDK provides an API to directly obtain the SDK version number, which you can call to get it.

```
OcrSDKKit.getInstance().getVersion()
```

SDK resource release

When your app exits or needs to reload the OCR feature, you can call the SDK resource release API.

```
@Override
protected void onDestroy() {
    if (mDialog != null) {
        mDialog.dismiss();
    }
    // Release OCR SDK resources
    OcrSDKKit.getInstance().release();
    super.onDestroy();
}
```

Configuring Obfuscation Rules

If the obfuscation feature is enabled for your app, add the following part to your obfuscation file to ensure the normal use of the SDK.

```
#Keep the custom OcrSDKKit class and class members from being obfuscated
-keep class com.tencent.ocr.sdk.** {*;}
third-party jar packages are not obfuscated
-keep class com.tencent.youtu.** {*;}
public library content is not obfuscated
-keep class com.tencent.could.** {*;}
```

FAQs

1. If integrating other SDKs simultaneously, the issue **More than one file was found with OS independent path 'lib/armeabi-v7a/libc++_shared.so'.** may occur.
   This is mainly because both the OCR SDK and other SDKs include the `libc++_shared.so` library. The solution is to add the following configuration in build.gradle:

```
android {
      ...
          // Filter the issue of repeated defining so
 packagingOptions{
     pickFirst 'lib/armeabi-v7a/libc++_shared.so'
 }
}
```

2. If the integrator uses the obfuscation tool AndResGuard, you can add the following obfuscation configuration:

```
// for OCR SDK
"R.string.ocr_*",
"R.string.rst_*",
"R.string.net_*",
"R.string.msg_*",
```

3. After integrating the OCR SDK, if the **Invoke-customs are only supported starting with Android O (--min-api 26)** error occurs?
You need to add the following configuration in build.gradle:

```
// java version supports 1.8
compileOptions {
 sourceCompatibility JavaVersion.VERSION_1_8
 targetCompatibility JavaVersion.VERSION_1_8
}
```

4. If you use the AutoSize component and also use the OCR's landscape mode, there may be an issue with abnormal page element sizes in landscape mode. The main reason is that AutoSize is set to the width and height baseline for portrait orientation by default. You can register the following callback in the Application to achieve adaptive baseline for portrait and landscape orientation:

```
  /**
* AutoSize adapts to the interface orientation, can be registered in
Application
* For example, design_width_in_dp 360, design_height_in_dp 640
(customers can modify it as they define)
*     <meta-data
```

```
*       android:name="design_width_in_dp"
*       android:value="360"/>
*      <meta-data
*       android:name="design_height_in_dp"
*       android:value="640"/>
*
*  DESIGN_WIDTH_DP = 360;
*  DESIGN_WIDTH_DP = 640;
*/
  public static void addAutoSizeListener() {
   AutoSizeConfig.getInstance().setOnAdaptListener(new
onAdaptListener() {
       @Override
       public void onAdaptBefore(Object target, Activity activity) {
           Context context = activity;
           int[] currentSize = ScreenUtils.getScreenSize(context);
           // Set the current screen size

AutoSizeConfig.getInstance().setScreenWidth(currentSize[0]);

AutoSizeConfig.getInstance().setScreenHeight(currentSize[1]);
           // Get the screen orientation of the current Activity
           int orientation =
activity.getResources().getConfiguration().orientation;
           // If it is in horizontal format, adjust the screen
reference
           if (orientation == Configuration.ORIENTATION_LANDSCAPE) {

AutoSizeConfig.getInstance().setDesignWidthInDp(DESIGN_HEIGHT_DP);

AutoSizeConfig.getInstance().setDesignHeightInDp(DESIGN_WIDTH_DP);
           } else {
               // If it is in vertical screen, adjust the screen
reference

AutoSizeConfig.getInstance().setDesignWidthInDp(DESIGN_WIDTH_DP);

AutoSizeConfig.getInstance().setDesignHeightInDp(DESIGN_HEIGHT_DP);
           }
       }

       @Override
       public void onAdaptAfter(Object target, Activity activity) {
```

```
        }
    });
}
```

Quick Integration (Harmony)

Last updated：2025-02-06 16:16:05

Development Preparation

1. Sign up for a Tencent Cloud account and click to enter the Optical Character Recognition Console to activate the service.
2. Obtain the API key in the account center.
3. Go to the Optical Character Recognition client SDK Integration page to download the corresponding SDK.

Access Process For Harmony OCR SDK

Introduction To Harmony OCR SDK

The file provided by the SDK is OCR_Harmony_Public_SDK_3.x.x.x.har (the specific version number is subject to the SDK delivery), which encapsulates the OCR recognition terminal capabilities. It currently includes identity card recognition, bank card recognition, business card recognition, vehicle Vin code recognition, license plate recognition, driving license recognition, and driving license recognition.

Environment Dependency

The current Harmony OCR recognition SDK is applicable to **API 11 (Harmony Next 4.0)** and above.

Access Steps

1. Add OCR_Harmony_Public_SDK_3.x.x.x.har to the libs directory in your project directory.

```
   ├── OcrHarmonyDemo
│     ├── build-profile.json5
│     ├── hvigorfile.ts
│     ├── libs
│     │     └── OCR_Harmony_Public_SDK_3.x.x.x.har  <<<==============
storage path for har package file
│     ├── oh-package.json5
│     └── src
│         └── main
```

2. Add the model file

```
   ├── OcrHarmonyDemo
│     ├── build-profile.json5
│     ├── hvigorfile.ts
│     ├── oh-package.json5
│     └── src
│         └── main
│             ├── module.json5
│             └── resources
│                 ├── rawfile
│                 │     ├── subject_common.iap  <<<============== store
the model file from the delivery package in this path, do not change
the file name
│                 └── zh_CN
│                     └── element
```

> **Note:**
> The model file is located in the delivery package under **sdk/models**. The model file includes a general model and a specific model for identity cards (the specific model for identity cards can detect both sides of the identity card). **The file name cannot be changed**. Choose the appropriate model as needed.

3. Modify the configuration in the oh-package.json5 of your project, as shown below:

```
{
  "license": "",
  "author": "",
  "name": "HuiyanHarmonyDemo",
  "description": "Please describe the basic information.",
```

```
  "main": "",
  "version": "1.0.0",
  "dependencies": {
    // Add Dependency
    "OcrPublicSDK": "file:./libs/OCR_Harmony_Public_SDK_3.x.x.x.har",
  }
}
```

SDK APIs Use Instructions

SDK initialization

Initialization is required before using the OCR SDK.

```
OcrSDKKit.getInstance().init(getContext());
```

The business modes currently supported by the OCR SDK are:

<OcrModeType Type>	Representation Of Meaning
OCR_DETECT_MANUAL	Manual shooting mode
OCR_DETECT_AUTO_MANUAL	Automatic Identification mode (Note: After 20s, prompt to switch to manual shooting)

Updating the temporary key

The OCR SDK supports using temporary key APIs. The main benefits of using temporary keys are: first, separating fixed keys from the terminal can enhance security; second, since the exchange of temporary keys is entirely under your control, you can control the API access permissions of end users based on custom rules. Therefore, it is recommended to use temporary keys. For details, refer to the documentation (Temporary Key Documentation and Process Link)

```
let fetch = new FetchToken()
fetch.fetchToken(false) // return temporary secretID, secretKey, token
```

OCR recognition (returns json string)

When you need to use the OCR recognition feature, you can directly call the recognition API for OCR business recognition.

```
// Start OCR recognition, recognition type is ID card front
let config: OcrSDKConfig = new OcrSDKConfigBuilder()
  .setOcrMode(this.ocrMode)
  .setAuto2ManualTimeout(this.timeout)
  .setCardType(ocrCardType)
  .setOcrUIConfig(uiConfig)
  .setSecretID(val[0])
  .setSecretKey(val[1])
  .setToken(val[2])
  .build();
OcrSDKKit.getInstance().startOcrByConfig(config, {
  onSuccess: (response: string, requestID: string, imgBase64: string) =>
{
    // successfully
  },
  onFail: (errorCode: number, errMsg: string) => {
    // fail
  }
});
```

SDK resource release

When your app exits or needs to reload the OCR feature, you can call the SDK resource release API.

```
OcrSDKKit.getInstance().release();
```

Client API Document Overview Of IOS APIs

Last updated: 2025-02-06 16:16:27

The main classes involved in the iOS Optical Character Recognition (OCR) SDK are OcrSDKKit, OcrSDKConfig, CustomConfigUI, and OcrCommmDef. The supported APIs are detailed below.

OcrSDKKit

OcrSDKKit is the external API class for OCR, with the main OCR features concentrated in this class.

API	Feature Description
sharedInstance()	Create a singleton of OcrSDKKit
clearInstance()	Proactively release resources
getVersion()	Obtain the current SDK version information
loadSDKConfig	Initialize SDK configuration information
updateFederationToken()	Proactively update the temporary key
startProcessOcr()	Start OCR recognition

SharedInstance()

```
+ (nonnull instancetype)sharedInstance;
```

- Feature description:
  Create a singleton of OcrSDKKit.
- Return result:
  The singleton object of OcrSDKKit.

ClearInstance()

```
/// clear SDK resources
+ (void)clearInstance;
```

- Feature description:
  A method to release resources proactively, which can be called when exiting the application or needing to clear resources.

GetVersion()

```
- (NSString *_Nonnull)getVersion;
```

- Feature description:
  Get the current version number of the SDK.
- Return result:
  The current version information of the SDK.

LoadSDKConfig()

```
/// SDKKit loads OCR configuration information. The key used here is a
fixed key. When using a temporary key, fill in nil for secretId and
secretKey.
/// @param secretId  Secret id
/// @param secretKey Secret key
/// @param ocrConfig OCR configuration class
- (int)loadSDKConfigWithSecretId:(NSString *)secretId withSecretKey:
(NSString *)secretKey withConfig:(OcrSDKConfig *)ocrSDKConfig;
```

- Feature description:
  Initialize SDK information.
- Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
OcrSDKConfig	ocrSDKConfig	Entity class for SDK configuration parameters
OcrSDKUIConfig	ocrSDKUIConfig	Entity class for SDK UI configuration parameters
NSString	secretId	SecretId key
NSString	secretKey	SecretKey key

UpdateFederationToken()

```
/// @param tmpSecretId temporary SecretId
/// @param tmpSecretKey temporary key information
/// @param token temporary exchange token
- (void)updateFederationToken:(NSString *_Nonnull) tmpSecretId
withTempSecretKey:(NSString *_Nullable)tmpSecretKey withToken:(NSString
*_Nonnull)token;
```

- Feature description:
  Proactively update temporary key information, call the setting proactively after you exchange with the server to obtain the temporary key.
- Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
NSString	tmpSecretId	Temporary key SecretId
NSString	tmpSecretKey	Temporary key SecretKey
NSString	token	Temporarily access token after exchange

StartProcessOcr()

```
/// Launch the SDK module, run the feature recognition module with a UI
interface
/// @param ocrType recognition mode
/// @param CustomConfigUI OCR UI configuration class. When nil is
passed, the default configuration is used.
/// @param onProcessSucceed success callback block
/// @param onProcessFailed failure callback block
- (void)startProcessOcr:(int)ocrType withSDKUIConfig:(CustomConfigUI
*)customConfigUI withProcessSucceedBlock:(OcrSDKKitProcessSucceedBlock
_Nonnull)onProcessSucceed withProcessFailedBlock:
(OcrSDKKitProcessFailedBlock _Nonnull)onProcessFailed;
```

- Feature description:
  Start OCR Recognition.
- Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
OcrType	ocrType	Type of OCR recognition started

CustomConfigUI	customConfigUI	ocrUI configuration class
OcrSDKKitProcessSucceedBlock	onProcessSucceed	The success callback of recognition
OcrSDKKitProcessFailedBlock	onProcessFailed	The failure callback of recognition

OcrSDKConfig

OcrSDKConfig is the configuration information entity of the SDK that needs to be passed in during OCR initialization.
Supported parameters and their default values are as follows:

Type	Name	Meaning	Default Value
OcrType	ocrType	Default recognition type	IDCardOCR_FRONT and IDCardOCR_BACK both represent id_card
OcrModeType	ocrModeType	Recognition mode types: OCR_DETECT_MANUAL represents manual capture mode, OCR_DETECT_AUTO_MANUAL represents automatic capture mode with a 20s prompt to switch to manual capture if no detection	OCR_DETECT_MANUAL default value
long	auto_timeout_ms	Timeout for automatically switching to manual capture mode in automatic capture mode (5000ms< `auto_timeout_ms` <180000ms)	20000ms
BOOL	cropIdCard	Enable ID card photo cropping (remove extra edges outside the document, automatically correct shooting angle) switch	NO

BOOL	cropPortrait	Enable portrait photo cropping (automatically extract the ID card portrait area)	NO
BOOL	copyWarn	Enable copy alarm	NO
BOOL	borderCheckWarn	Enable border and in-frame occlusion alarm	NO
BOOL	reshootWarn	Enable re-photographing alarm	NO
BOOL	detectPsWarn	Enable PS detection alarm	NO
BOOL	tempIdWarn	Enable temporary identity card alarm	NO
BOOL	invalidDateWarn	Enable invalid identity card validity date alarm	NO
BOOL	quality	Enable image quality score (evaluate image blurriness)	NO
BOOL	multiCardDetect	Enable multi-card detection or not	NO
BOOL	reflectWarn	Enable reflection alarm or not	NO
NSString	retImageType	Image preprocessing, detect the skew angle of the image, rotate the originally skewed image around the center point to correct it, and finally output a corrected business card cutout.	Null
BOOL	retImage	Whether the Malaysia identity card API returns an image	NO
BOOL	retBorderCutImage	Whether to return preprocessed (precisely cropped and aligned) **bank**	NO

		card image data (only effective under BankCardOCR type)	
BOOL	retCardNoImage	Whether to return the cropped image data of the **bank card** number (only effective under **BankCardOCR** type)	NO
BOOL	enableCopyCheck	**Bank card** copy detection switch, if the input image is a **bank card** copy image, an alarm will be returned (only effective under **BankCardOCR** type)	NO
BOOL	enableReshootCheck	**Bank card** rephotography detection switch, if the input image is a **bank card** rephotography image, an alarm will be returned (only effective under **BankCardOCR** type)	NO
BOOL	enableBorderCheck	**Bank card** border occlusion detection switch, if the input image has the **bank card** border occluded, an alarm will be returned (only effective under **BankCardOCR** type)	NO
BOOL	enableQualityValue	Whether to return the **bank card** image quality score (the image quality score is a standard for evaluating the blurriness of an image) (only effective under **BankCardOCR** type)	NO
BOOL	detectFake	Whether to verify the authenticity of the **Hong Kong ID card** (only effective under HKIDCardOCR_03 and HKIDCardOCR_18)	NO

BOOL	returnHeadImage	Whether to return the portrait photo of the **Hong Kong ID card** (only effective under **HKIDCardOCR_03 and HKIDCardOCR_18**)	NO

CustomConfigUI

CustomConfigUI is the UI configuration entity class of the SDK that needs to be passed in when starting the SDK module.
Supported parameters and their default values are as follows:

Type	Name	Meaning	Default Value
BOOL	isShowTips	Whether to display the middle note	YES
NSString	tipsShowText	Middle note content (limited to 15 Chinese characters)	"Please avoid content with folded corners, coverage, and reflection"
NSString	remindDialogText	Dialog mode switch note text	"Failed to recognize the document, switch to take photo and upload?"
UIColor	remindConfirmColor	Dialog mode switch note, button color	RGBA: 5 106 1 1
UIColor	cardFrameColor	Color of the selected card frame	RGBA: 5 106 1 1
UIImage	takePictureImage	Photo button icon 80x80	Default icon
UIImage	lightONImage	Flashlight button icon 40x40	Default icon
UIImage	lightOFFImage	Flashlight off button icon 40x40	Default icon
UIImage	albumImage	Album button icon 40x40	Default icon
BOOL	isShowAlbumBtn	Whether to show album button	YES

BOOL	isHorizontal	Display in landscape mode	NO
float	blurThreshold	Minimum range of light blur	0.5
int	validCount	Number of stable frames	4
OCR_LanguageType	languageType	Set current language	OCR_DEFAULT
OcrCameraZoom	zoomType	Set default zoom factor	OCR_ZOOM_1X
BOOL	openClipImage	Enable image cropping	YES
NSString	ocrSDKBundlePath	Set OCRSDK bundle file path, default is read from main bundle	nil

OcrType

OcrType is an enumeration type that lists the business types supported by the current OCR SDK, roughly as follows:

OcrType Type	Representation Of Meaning
OcrType.IDCardOCR_FRONT	ID card portrait side recognition mode
OcrType.IDCardOCR_BACK	ID card national emblem side recognition mode
OcrType.BankCardOCR	Bank card front recognition mode
OcrType.BusinessCardOCR	Business card front recognition mode
OcrType.MLIdCardOCR	Malaysian ID card recognition mode
OcrType.LicensePlateOCR	Vehicle license plate recognition mode
OcrType.VinOCR	Vehicle VIN code recognition mode

OcrType.VehicleLicenseOCR_FRONT	Driving license home page recognition mode
OcrType.VehicleLicenseOCR_BACK	Driving license secondary page recognition mode
OcrType.DriverLicenseOCR_FRONT	Driving license home page recognition mode
OcrType.DriverLicenseOCR_BACK	Driving license secondary page recognition mode
OcrType.VinOCR2	Vehicle VIN code recognition mode (high-precision version)
OcrType.HKIDCardoCR_03	Hong Kong (China) identity card version 03 recognition mode
OcrType.IDCardOCR_HK18	Hong Kong (China) identity card version 18 recognition mode
OcrType.PermitOCR	Hong Kong (China), Macao (China), and Taiwan (China) travel permit recognition mode
OcrType.MLIDPassportocR	International passport recognition mode
OcrType.HmtResidentPermitOCR	Residence permit for Hong Kong (China), Macao (China), and Taiwan (China)

OcrModeType

OcrModeType is an enumeration type that lists card recognition modes.

<OcrModeType Type>	Representing Meaning
OCR_DETECT_MANUAL	Manual shooting mode
OCR_DETECT_AUTO_MANUAL	Automatic identification mode (Note: After 20s, prompt whether to switch to manual shoot)

OCR_LanguageType

OCR_LanguageType is an enumeration type that lists language settings.

Type	Representing Meaning
OCR_DEFAULT	Follow the system
OCR_ZH_HANS	Simplified Chinese (Simplified Chinese)
OCR_ZH_HANT	Traditional Chinese
OCR_EN	English

OcrCameraZoom

OcrCameraZoom is an enumeration type that lists the default camera zoom levels.

Type	Representing Meaning
OCR_ZOOM_1X	1x
OCR_ZOOM_1_5X	1.5x
OCR_ZOOM_2X	2x

Callback For Recognition Result

Callback class for OCR recognition results, used to receive recognition results and error exceptions.

```
/// SDKKit success handling callback interface
/// @param resultInfo will return the success message based on different
working modes (usually a Network Packet Return JSON field)
/// @param reserved Reserved
typedef void (^OcrSDKKitProcessSucceedBlock)(id _Nonnull resultInfo,
UIImage *resultImage,id _Nonnull reserved);

/// SDKKit failure handling callback interface
/// @param error Exception error triggered during the process
/// @param reserved Reserved
///tips
typedef void (^OcrSDKKitProcessFailedBlock)(NSError *_Nonnull error, id
_Nullable reserved);
```

Note

User cancellation of OCR exit will trigger the OcrSDKKitProcessFailedBlock callback

- domain: "OcrSdk.UserCancelOcr" – code: 200101
- NSLocalizedDescription: "User manually stopped OCR"

Example of resultInfo for identity card front side request return:

```
{
    "Name": "Li Ming"
    "Sex": "Male",
    "Nation": "Han",
    "Birth": "1987/1/1",
    "Address": "Tencent Building, High-tech Park, Shijingshan District,
Beijing",
    "IdNum": "440524******010014",
    "Authority": "",
    "ValidDate": "",
    "AdvancedInfo": "{}",
    "RequestId": "ab2c132e-9e1c-43d3-b0ef-9b4d80f00330"
 }
```

Example of resultInfo for identity card back side request return:

```
{
    "Name": "",
    "Sex": "",
    "Nation": "",
    "Birth": "",
    "Address": "",
    "IdNum": "",
    "Authority": "Zhaoxian Public Security Bureau",
    "ValidDate": "2010.07.21-2020.07.21",
    "AdvancedInfo": "{}",
    "RequestId": "0d394478-6d4d-48fc-8b19-552415bf46de"
 }
```

Example of resultInfo for bank card request return:

```
{
    "CardNo": "6225760088888888",
    "BankInfo": "China Merchants Bank (03080000)",
    "ValidDate": "08/2022",
    "RequestId": "46ab2e62-11e3-4d04-9fab-0abe18e7c927"
```

```
  }
```

Example of resultInfo for business card request result return:

```
{
    "BusinessCardInfos": [
      {
        "Name": "name",
        "Value": "Amy"
      },
      {
        "Name": "position",
        "Value": "Visual Designer"
      },
      {
        "Name": "department",
        "Value": "Social Platform Department"
      },
      {
        "Name": "company"
        "Value": "Tencent"
      },
      {
        "Name": "address"
        "Value": "Tencent Building, Keji Zhongyi Road, High-tech Park,
Nanshan District, Shenzhen"
      },
      {
        "Name": "mailbox",
        "Value": "ab***fg@tencent.com"
      },
      {
        "Name": "mobile phone",
        "Value": "+86-133****5678"
      },
      {
        "Name": "QQ",
        "Value": "1234567"
      },
      {
        "Name": "WeChat",
        "Value": "amy001"
      }
```

```
    ],
    "RetImageBase64": "",
    "RequestId": "98f8fcbf-933a-4e95-ac48-6f1a9308fs6h"
  }
```

Example of resultInfo for Malaysian identity card request result return:

```
{
  "Response": {
    "Name": "KAVIN ONG KHI MN",
    "ID": "710716-08-6085",
    "Address": "NO 11 PERSIARN PERAJRIT 4 TAMA PERAK 31400 IPOH ERAK",
    "Sex": "LEAKI",
    "Birthday": "",
    "Warn": [],
    "Image": "",
    "AdvancedInfo": "{\"ID\":{\"Confidence\":\"1.0000\"},\"Name\":
{\"Confidence\":\"0.9996\"},\"Address\":
{\"Confidence\":\"0.9997\"},\"Sex\":{\"Confidence\":\"0.9999\"}}",
    "Type": "MyKad",
    "RequestId": "c969da05-54e3-4d0a-a55d-b3ef90d4ebf5"
  }
}
```

Example of resultInfo for license plate recognition request result return:

```
{
  "Response": {
    "Number": "Jing N0L9U8",
    "Confidence": 99,
    "Rect": {
      "X": 217,
      "Y": 233,
      "Width": 170,
      "Height": 21
    },
    "RequestId": "210103d3-db06-4691-abe0-c0853aae606b"
  }
}
```

Example of resultInfo for vehicle VIN code recognition request result return:

```
{
  "Response": {
    "Vin": "LBV2B25G2E5069977",
    "RequestId": "c59d9002-6c8c-426d-b57f-a8837dee2c7c"
  }
}
```


Example of resultInfo for driving license home page and secondary page request result return:


```
{
  "Response": {
    "FrontInfo": {
      "PlateNo": "Hu AA1234",
      "VehicleType": "Compact Car",
      "Owner": "Li Ming",
      "Address": "6F, Tengyun Building, 397 Tianlin Road, Xuhui
District, Shanghai",
      "UseCharacter": "Non-commercial",
      "Model": "Buick SGM7151LAAA",
      "Vin": "ABCDEFGH123456789",
      "EngineNo": "8B54321",
      "RegisterDate": "2011-10-10",
      "IssueDate": "",
      "Seal": "Shanghai Traffic Police Brigade, Shanghai Public Security
Bureau"
    },
    "BackInfo": null,
    "RecognizeWarnCode": [
      -9106
    ],
    "RecognizeWarnMsg": [
      "WARN_DRIVER_LICENSE_PS_CARD"
    ],
    "RequestId": "820916b4-b391-40a8-9203-7ae87e3f1954"
  }
}
```


Example of resultInfo for driving license home page and secondary page request result return:


```
{
  "Response": {
```

```
    "Name": "Li Ming"
    "Sex": "Male",
    "Nationality": "China"
    "Address": "6F, Tengyun Building, 397 Tianlin Road, Xuhui District,
Shanghai"
    "DateOfBirth": "1987-01-01",
    "IssuingAuthority": "Shanghai Traffic Police Brigade, Shanghai
Public Security Bureau"
    "DateOfFirstIssue": "2011-10-01",
    "Class": "C1",
    "StartDate": "2011-10-01",
    "EndDate": "2017-10-01",
    "CardCode": "440524198701010014",
    "ArchivesCode": "",
    "Record": "",
    "RecognizeWarnCode": [
      -9106
    ],
    "RecognizeWarnMsg": [
      "WARN_DRIVER_LICENSE_PS_CARD"
    ],
    "RequestId": "4ba2958b-e7cf-41c2-aafe-fdc985307f63"
  }
}
```


For the returned error code and error information, please refer to Error Code.


```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated.
Please check your signature is correct."
        },
      "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

Overview Of Android APIs

Last updated: 2025-02-06 16:16:40

The main classes involved in the Android OCR SDK are OcrSDKKit, OcrSDKConfig, OcrType, OcrModeType, ISDKKitResultListener, ISdkOcrEntityResultListener, and OCR recognition return entity class. The supported APIs are detailed below.

OcrSDKKit

OcrSDKKit is the external API class for OCR, with the main OCR features concentrated in this class.

API	Feature Description
getInstance()	Create a singleton of OcrSDKKit
release()	Proactively release resources
getVersion()	Obtain the current SDK version information
initWithConfig()	Initialize SDK configuration information
updateFederationToken()	Proactively update the temporary key
startProcessOcr()	Start OCR recognition, return JsonString
startProcessOcrResultEntity()	Start OCR recognition, return entity objects corresponding to different modes

getInstance()

```
public static OcrSDKKit getInstance()
```

- Feature description:
  Create a singleton of OcrSDKKit.
- Return result:
  Singleton object of OcrSDKKit.

release()

```
public void release()
```

- Feature description:
  A method to release resources proactively, which can be called when exiting the application or needing to clear resources.

getVersion()

```
public final String getVersion()
```

- Feature description:
  Get the current version number of the SDK.
- Return result:
  The current version information of the SDK.

initWithConfig()

```
public void initWithConfig(Context context, OcrSDKConfig config)
```

- Feature description:
  Initialize SDK information.
- Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
Context	context	Application context information
OcrSDKConfig	config	Entity class for SDK configuration parameters

updateFederationToken()

```
public void updateFederationToken(final String tmpSecretId, final String
tmpSecretKey,final String token)
```

- Feature description:
  Proactively update temporary key information, call the setting proactively after you exchange with the server to obtain the temporary key.
- Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
String	tmpSecretId	Temporary key SecretId

String	tmpSecretKey	Temporary key SecretKey
String	token	Temporarily access token after exchange

startProcessOcr()

```
public void startProcessOcr(Activity activity, OcrType ocrType,
CustomConfigUi customConfigUi, ISDKKitResultListener resultListener)
```

- Feature description:
  Launch the default interface for OCR recognition, with customizable UI elements. (**If CustomConfigUi is passed as null, the default UI configuration will be used**)
- Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
Activity	activity	Current Activity object of the default OCR interface
OcrType	ocrType	Type of OCR recognition started
CustomConfigUi	customConfigUi	Interface configuration parameters passed when launching the default interface. To use the default configuration, pass null.
ISDKKitResultListener	resultListener	Callback object for receiving OCR recognition results

startProcessOcrResultEntity()

```
public <T extends OcrResult> void startProcessOcrResultEntity(Activity
activity, OcrType ocrType,CustomConfigUi customConfigUi, Class<T>
entity,ISdkOcrEntityResultListener<T> ocrEntityResultListener)
```

- Feature description:
  Launch the default interface for OCR recognition, with customizable UI elements. (**If CustomConfigUi is passed as null, the default UI configuration will be used**), and return different recognition result types based on the specified recognition mode.
- Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
Activity	activity	Current Activity object of the default OCR interface
OcrType	ocrType	Type of OCR recognition started
CustomConfigUi	customConfigUi	Interface configuration parameters passed when launching the default interface. To use the default configuration, pass null.
Class < T >	entity	Entity class for OCR recognition results
ISdkOcrEntityResultListener	ocrEntityResultListener	Callback object for receiving OCR recognition results
< T > extends OcrResult	< T >	Sub-type of OCR recognition result OcrResult

Entity class of OCR recognition result, sub-type of OCR recognition result OcrResult.

OcrSDKConfig

OcrSDKConfig is the configuration information entity class of the SDK that needs to be passed in during OCR initialization, using the builder method for parameter configuration. Supported parameters and their default values are as follows:

Type	Name	Meaning	Default Value
OcrType	ocrType	Default recognition type	IDCardOCR_FRONT and IDCardOCR_BACK both represent id_card
int	cardType	In identity card mode, 0 is front, 1 is back	0 front
int	modeType	Recognition mode type: 0 represents manual shoot mode, 1 represents automatic capture mode, 2 represents automatic + manual mode (automatic first, then manual shoot after timeout)	2 represents automatic + manual mode
int	autoTimeout	Automatic capture timeout (in milliseconds, minimum 5 seconds, internal limit 30 seconds)	20000 ms

String	resultUrl	ResultUrl information of the recognition request	`https://ocr.tencentcloudapi.com/`
String	secretId	Key information used for the request (if using fixed key mode, a fixed key can be passed in)	Null
String	secretKey	Key information used for the request (if using fixed key mode, a fixed key can be passed in)	Null
String	tempToken	Temporary token information used for the request	Null
boolean	cropIdCard	Enable ID card photo cropping (remove extra edges outside the document, automatically correct shooting angle) switch	false
boolean	cropPortrait	Enable portrait photo cropping (automatically extract the ID card portrait area)	false
boolean	copyWarn	Enable copy alarm	false
boolean	borderCheckWarn	Enable border and in-frame occlusion alarm	false
boolean	reshootWarn	Enable re-photographing alarm	false
boolean	detectPsWarn	Enable PS detection alarm	false
boolean	tempIdWarn	Enable temporary identity card alarm	false
boolean	invalidDateWarn	Enable invalid identity card validity date alarm	false
boolean	quality	Enable image quality score (evaluate image blurriness)	false

String	retImageType	Image preprocessing, detect the skew angle of the image, rotate the originally skewed image around the center point to correct it, and finally output a corrected business card cutout.	Null
boolean	canCancelWaiting	Allow active cancellation by clicking back during the identification process (when a loading prompt box appears)	false
boolean	isOpenClipImage	Whether to enable the capture feature when selecting an image	true
boolean	retBorderCutImage	For bank cards, whether to return the preprocessed (precisely cropped and aligned) bank card image data	false
boolean	retCardNoImage	For bank cards, whether to return the cropped image data of the card number, default is false.	false
boolean	enableQualityValue	For bank cards, whether to return the image quality score (the image quality score is a standard for evaluating the blurriness of an image)	false

OcrType

OcrType is an enumeration type that lists the business types supported by the current OCR SDK, roughly as follows:

<OcrType Type>	Representing Meaning	Corresponding Result Entity Class
OcrType.IDCardOCR_FRONT	ID card portrait side recognition mode	IdCardOcrResult
OcrType.IDCardOCR_BACK	ID card national emblem side recognition mode	IdCardOcrResult
OcrType.BankCardOCR	Bank card front recognition mode	BankCardOcrResult

OcrType.BusinessCardOCR	Business card front recognition mode	BusinessCardOcrResult
OcrType.VinOCR	Vehicle VIN recognition mode	VinOcrResult
OcrType.LicensePlateOCR	Vehicle license plate recognition mode	CarLicensePlateResult
OcrType.DriverLicenseOCR_FRONT	Driving license home page recognition mode	DriverLicenseCardResult
OcrType.DriverLicenseOCR_BACK	Driving license secondary page recognition mode	DriverLicenseCardResult
OcrType.VehicleLicenseOCR_FRONT	Driving license home page recognition mode	VehicleLicenseCardResult
OcrType.VehicleLicenseOCR_BACK	Driving license secondary page recognition mode	VehicleLicenseCardResult
OcrType.GENERAL_VIN	General vehicle vin code recognition mode (mainly recommended for photo mode)	VinOcrResult
OcrType.IDCardOCR_HK03	Hong Kong (China) identity card version 03 recognition mode	HKIDCardOcrResult
OcrType.IDCardOCR_HK18	Hong Kong (China) identity card version 18 recognition mode	HKIDCardOcrResult
OcrType.Exit_Entry_HK_Macao_Card	Hong Kong (China), Macao (China), and Taiwan (China) travel permit recognition mode	PermitOcrResult
OcrType.MLID_PASSPORT	International passport recognition mode	MLIDPassportOcrResult
OcrType.HMT_RESIDENT_PERMIT_OCR	Residence permit for Hong Kong (China), Macao (China), and Taiwan (China)	HmtResidentPermitOcrResult

OcrModeType

OcrModeType is an enumeration type that lists card recognition modes.

<OcrModeType Type>	Representation Of Meaning

OCR_DETECT_MANUAL	Manual shooting mode
OCR_DETECT_AUTO_MANUAL	Automatic identification mode (Note: After 20s, prompt whether to switch to manual shoot)

LanguageStyle

LanguageStyle is an enumeration type that lists supported language types.

LanguageStyle Type	Representation Of Meaning
AUTO	Follow system language
ENGLISH	English
SIMPLIFIED_CHINESE	Simplified Chinese

CustomConfigUi

This is a configuration class for user-customized UI. The currently supported properties are shown in the table below and can be set via javabean set.

Type	Name	Meaning
boolean	isShowTitleBar	Set whether to display the default interface's title bar
String	titleBarText	Set the text content displayed in the title bar
int	titleColor	Set the title bar background color (0xFFFFFF type)
String	remindDialogText	Set the reminder dialog text content
int	remindDialogTextColor	Set the reminder dialog text color (0xFFFFFF type)
int	remindDialogTextSize	Set the reminder dialog text size
int	remindDialogConfirmColor	Set the color of the confirmation button (switch mode) in the reminder dialog (0xFFFFFF type)

String	remindDialogConfirmText	Set the text content of the confirmation button in the reminder dialog
int	remindDialogCancelColor	Set the color of the cancel button in the reminder dialog (0xFFFFFF type)
String	remindDialogCancelText	Set the text content of the cancel button in the reminder dialog
boolean	remindDialogShowTitle	Whether to display the title of the Note dialog
int	cardFrameColor	Set the color of the four corners when the card preview window is selected (0xFFFFFF type)
int	successRemindTextColor	Set the text color of the prompt after custom capture or shooting success (0xFFFFFF type)
Int	statusBarColor	Set the status bar background color (0xFFFFFF type)
boolean	isShowStatusBar	Set whether to display the status bar in the default interface, default is true
int	imageSelectResId	Set the icon resource id for selecting images from the image library in the default interface
int	lightImageOnResId	Set the icon resource id for turning on the flash in the default interface
int	lightImageOffResId	Set the icon resource id for turning off the flash in the default interface
int	takePicturesResId	Set the icon resource id for the active photo button in the default interface
int	backActionIconResId	Set the icon resource id for the back button in the default interface
boolean	isRemoveAlbum	Set whether to display the album selection button in the default interface, default is to display
boolean	isRemoveFlash	Set whether to display the flash click button in the default interface, default is to display

boolean	isLandscape	Set the default interface to landscape mode, default is false
boolean	isShowTips	Set whether to display the tips box in the default interface, default is true
String	showTipsText	Set the reminder tips text content, default is "Please avoid folded corners, coverage, and reflection" (no more than 15 words)
boolean	isShowStatusBar	Set whether to display the status bar in the default interface, default is true
int	showTipsTextColor	Set the reminder tips text color (0xFFFFFF type)
int	showTipsTextSize	Set the reminder tips text size
boolean	isShowTipsIcon	Set whether to display the icon on the reminder tips
boolean	isShowTipsBackground	Set whether to display the background box of the reminder tips
int	portraitLayoutResId	Resource ID for portrait layout, refer to custom UI layout for specific usage
int	landscapeLayoutResId	Resource ID for landscape layout, refer to custom UI layout for specific usage
boolean	useDeepColorStatusBarIcon	Whether to use dark status bar icons and text (used with light color status bar), default is false
String	bottomTipsContext	Dynamic setting of the content of the tips below the preview window **("\n", can actively newline)**

Custom UI layout

OCR SDK supports custom UI layout. The SDK provides default vertical layout file **demo_ocr_detect_fragment.xml** and landscape layout file

demo_ocr_detect_hor_fragment.xml. Integrators can modify the UI based on these files. **Do not delete or modify the default views and their corresponding IDs.**
Related resources can be found in the **res folder** within the installation package provided by the SDK.
Below is a default vertical layout file:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <!--  Preview view of the camera -->
    <TextureView
        android:id="@+id/camera_surface_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <!--  Overall overlay view and recognition frame -->
    <com.tencent.ocr.sdk.component.CameraMaskView
        android:id="@+id/ocr_mask_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:txy_line_color="@color/blue"
        app:txy_line_length="30dp"
        app:txy_line_margin="5dp"
        app:txy_line_padding="0dp"
        app:txy_line_width="3dp"
        app:txy_maskView_view_type="id_back"
        app:txy_mask_color="@color/grey_text"
        app:txy_mask_margin="90dp"
        app:txy_position_flag="margin"
        app:txy_tip_color="@color/white"
        app:txy_tip_margin="20dp"
        app:txy_tip_size="15sp" />

    <!--  Custom tips display interface -->
    <com.tencent.ocr.sdk.component.OcrDetectTipsView
        android:id="@+id/ocr_tips_tv"
        android:layout_width="wrap_content"
        android:layout_height="40dp"
```

```
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="370dp" />

    <!--  Interface for the three buttons at the bottom -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="30dp"
        android:layout_marginEnd="30dp"
        android:layout_marginBottom="26dp"
        android:gravity="center_vertical|center_horizontal"
        android:orientation="horizontal">
        <!--  Album selection button -->
        <ImageButton
            android:id="@+id/album_image_button"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:layout_marginRight="50dp"
            android:background="@drawable/demo_txy_ocr_photo_album" />

        <!--  Take photo button  -->
        <ImageButton
            android:id="@+id/take_picture_button"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:background="@drawable/demo_txy_ocr_take_pictures" />

        <!-- Flash button  -->
        <ImageButton
            android:id="@+id/light_image_button"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:layout_marginStart="50dp"
            android:background="@drawable/demo_txy_ocr_light_off" />
    </LinearLayout>

    <!-- View for the default back button  -->
    <ImageView
        android:visibility="invisible"
        android:id="@+id/txy_detect_back"
        android:layout_marginTop="16dp"
```

```
        android:layout_marginStart="10dp"
        android:src="@drawable/txy_action_back"
        android:layout_width="32dp"
        android:layout_height="32dp"/>

    <!-- Upper title layout  -->
    <include layout="@layout/demo_show_title_bar"/>
</RelativeLayout>
```

If you add a view to the default layout and need to control the view, you can call and register it as follows:

```
OcrSDKKit.getInstance().setOcrEventListener(new OcrEventListener() {
    @Override
    public void onMainViewCreate(View mainView) {
        // Callback for interface creation (main thread)
        if (mainView != null) {
            Button button =
mainView.findViewById(R.id.demo_test_button);
            // Registration and handling of button events, be sure to
check null for View!
            button.setOnClickListener(v -> {
               Log.e(TAG, "click");
            });
        }
    }

    @Override
    public void onMainViewDestroy() {
        // Callback for interface destruction (main thread)
    }
});
```

Modify CameraMaskView

CameraMaskView is the mask component of the subject, providing modification through attribute configuration, which can be directly operated in XML. The currently supported attributes are as follows:

Attribute	Type	Meaning
txy_mask_color	color|referen	Default color of the outer mask

	ce	
txy_tip_text	string|reference	Default reminder text
txy_tip_color	color|reference	Color of the upper reminder text (default)
txy_tip_light_color	color|reference	Color of the upper reminder text (correct reminder)
txy_tip_error_color	color|reference	Color of the upper reminder text (alert)
txy_tip_size	integer|dimension	Size of the upper reminder text
txy_tip_margin	integer|dimension	Distance between the upper reminder text and the select box
txy_tip_textStyle	enum[normal, bold, italic, bold_italic]	Font style of the upper reminder text
txy_tip_show	boolean	Whether to display the upper reminder, default is display
txy_line_color	color|reference	Border line color of the select box (default)
txy_line_light_color	color|reference	Border line color of the select box (correct reminder)
txy_line_error_color	color|reference	Border line color of the select box (alert)
txy_line_width	integer|dimension	Width of the select box border line
txy_line_length	integer|dimension	Length of the select box border line
txy_line_padding	integer|dimension	Distance from the select box border line to the central transparent box
txy_line_margin	integer|dimension	Distance from the select box border line to the mobile screen

txy_position_flag	enum[center, margin]	Position of the select box, center means centered, margin means top
txy_mask_margin	integer|dimension	txy_position_flag = margin, position of the select box from the top
txy_mask_view_use_type	enum[portrait, landscape]	portrait means adapted to vertical format, landscape adapted to horizontal format
txy_rect_height_weight_hor	integer	Only applicable to horizontal format, proportion of the middle height occupying the screen
txy_front_head_icon	reference	Replacement image of the front portrait @drawable/head
txy_back_emblem_icon	reference	Replacement image of the back emblem @drawable/emblem
txy_bottom_tips_textStyle	enum[normal, bold, italic, bold_italic]	Font style of the lower reminder text
txy_bottom_tips_txt	string|reference	Default text content of the lower reminder, **("\n", can actively newline)**
txy_bottom_tip_size	integer|dimension	Font size of the lower reminder text
txy_bottom_tip_color	color|reference	Color of the lower reminder text
txy_bottom_tip_margin_top	integer|dimension	Distance between the lower reminder and the upper edge of the box
txy_bottom_tip_show	boolean	Whether to display the lower reminder, default is not to display
txy_is_show_scanline	boolean	Whether to display the built-in scan line, default is not display
txy_scanline_animator_time	integer	Time required for the scan line to complete one scan animation **(Unit: ms)**, default is 1500ms
txy_scanline_start_color	color|reference	Starting color of the scan line animation area, 0-40% of the area is this color
txy_scanline_mid_color	color|reference	Transition color of the scan line animation area, 40%-80% is a gradient effect from the start color

		to this color
txy_scanline_end_color	color|reference	End color of the scan line animation area, 80%-100% is a gradient effect from the transition color to this color
txy_scanline_start_color_size_range	integer	The percentage from 0 to this position is the start color, default is 40 (percentage)
txy_scanline_start_to_mid_color_size_range	integer	The range from txy_scanline_start_color_size_range to this position is the gradient area from start to transition color, default is 80 (percentage)

```
<!-- Reference scan line color in the demo -->
<!-- Start color -->
<color name="txy_scanline_start_color">#00000000</color>
<!-- Transition color -->
<color name="txy_scanline_mid_color">#22F77C61</color>
<!-- End color -->
<color name="txy_scanline_end_color">#EEF77C61</color>
```

If there is a conflict between the above and CustomConfigUi passed attributes, the configuration in CustomConfigUi has a higher priority.

Custom Tips and Some UI Text

The SDK will provide **demo_tips_string.xml**. If you need to modify the tips, you can keep the key unchanged and directly modify the corresponding value. Finally, place demo_tips_string.xml under the resources of the module that depends on the OCR SDK for packaging. Android Studio will automatically complete the resource replacement and overwrite. The currently supported replacement content is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Default information on the dialog (接口名) -->
    <string name="txy_ocr_cancel">cancel</string>
    <string name="txy_ocr_ok">confirm</string>
    <string name="txy_ocr_change_mode">Switch mode</string>
    <string name="txy_ocr_tips">Note</string>
    <string name="txy_ocr_change_mode_info">Content not recognized,
switch mode to upload photo?</string>
```

```
    <string name="txy_ocr_loading">Data loading</string>
    <string name="txy_ocr_back">Back</string>
    <string name="txy_ocr_title">Title</string>
    <string name="txy_ocr_setting">Settings</string>
    <string name="txy_ocr_permission_info">You have denied or disabled
permission, please manually grant access</string>
    <string name="txy_ocr_change_mode_manual">Switched to manual
shoot</string>
    <string name="txt_user_cancel_ocr">User actively stopped
OCR</string>
    <string name="txt_inner_sdk_ocr">OCR internal error, please
retry</string>
    <string name="txt_select_image_error">An exception occurred while
selecting image</string>
    <string name="txy_auto_mode_time_out_error">Failed to automatically
capture photo</string>
    <string name="txy_ocr_tip_default_txt">Please avoid folded corners,
coverage, and reflective surfaces in the content to be
recognized</string>
    <!-- Text information in OCR -->
    <string name="txy_ocr_tip_please">Please place</string>
    <string name="txy_ocr_tip_id_card">Identity card</string>
    <string name="txy_ocr_tip_bank_card">Bank card</string>
    <string name="txy_ocr_tip_business_card">Business card</string>
    <string name="txy_ocr_tip_card">Card</string>
    <string name="txy_ocr_tip_car_card">License plate</string>
    <string name="txy_ocr_tip_car_vin">Vehicle vin code</string>
    <string name="txy_ocr_tip_driver_license">Driving license</string>
    <string name="txy_ocr_tip_vehicle_license">Driving license</string>
    <string name="txy_ocr_tip_hold_in_kuang">Place within frame</string>
    <string name="txy_ocr_txt_identify">Recognition</string>
    <string name="txy_ocr_hk_idcard_03">Hong Kong Identity Card
03</string>
    <string name="txy_ocr_hk_idcard_18">Hong Kong Identity Card
18</string>
    <string name="txy_ocr_exit_entry_hk_macao_card">Hong Kong and Macao
Travel Permit</string>
    <!-- Error message -->
    <string name="txy_ocr_error_result_txt">OCR recognition request
failed</string>
    <!-- Error message -->
    <string name="txy_ai_camera_open_error">Failed to open
camera</string>
```

```
    <string name="txy_ai_camera_close_error">Failed to close
camera</string>
    <string name="txy_ai_camera_preview_error">Failed to preview
camera</string>

    <!-- Note -->
    <string name="ocr_auto_succeed">Automatic capture successful, please
wait for the verification result</string>
    <string name="ocr_manual_succeed">Manual capture successful, please
wait for the verification result</string>
    <string name="ocr_card_closer">Please get closer</string>
    <string name="ocr_card_farer">Please move farther</string>
    <string name="ocr_cam_blur">Blurred focus</string>
    <string name="ocr_no_card">Please insert</string>
    <string name="ocr_pose_keep">Please keep the content within
frame</string>
    <string name="ocr_auto_timeout">Automatic capture timeout</string>
    <string name="ocr_switch_to_auto">Automatic capture</string>
    <string name="ocr_switch_to_manual">Manual capture mode, please
click the photo button</string>
    <string name="ocr_cam_begin_focus">Auto adjust focal length</string>
    <string name="ocr_cam_end_focus">Focal length adjustment
complete</string>
</resources>
```

IdCardOcrResult

Result entity object of ID card OCR recognition

Type	Name	Meaning
String	name	Name
String	sex	Gender
String	nation	Ethnicity
String	birth	Date of Birth
String	address	Address
String	idNum	ID No.
String	authority	Issuing authority

String	validDate	Valid Date
String	requestId	Request ID
String	advancedInfo	Extension field content Refer to the official website

BankCardOcrResult

Result entity object of bank card OCR recognition

Type	Name	Meaning
String	bankInfo	Bank information
String	cardNo	Bank card number
String	validDate	Valid Date
String	requestId	Request ID
String	cardType	Bank card type
String	cardName	Bank card name
String	borderCutImage	Slice image data, null if not available
String	cardNoImage	Card number image data, null if not available

BusinessCardOcrResult

The result entity object of business card OCR recognition, mainly contains a series of BusinessCardItems objects.

Type	Name	Meaning
List< BusinessCardItems >	businessCardInfos	List of business card elements
String	requestId	Request ID

BusinessCardItems

Basic elements of business card OCR recognition, including name and information.

Type	Name	Meaning

String	name	Sub-item Name
String	value	Sub-item content

VinOcrResult

Result entity object of vehicle VIN recognition

Type	Name	Meaning
String	vin	Vehicle VIN code

CarLicensePlateResult

Type	Name	Meaning
String	number	License plate string information
int	confidence	Confidence degree of this recognition
OcrRect	rect	Pixel coordinate box of the text line in the original image

OcrRect

Rectangle coordinates

Type	Name	Meaning
int	x	Coordinate x of the top-left corner
int	y	Coordinate y of the top-left corner
int	width	Width
int	height	Height

DriverLicenseCardResult

Result entity class of driving license recognition

Type	Name	Meaning
String	name	Name
String	sex	Gender

String	nationality	Nationality
String	address	Address
String	dateOfBirth	Date of Birth
String	dateOfFirstIssue	Date of first license issuance
String	classType	Permitted vehicle type
String	startDate	Valid period start time
String	endDate	Valid period deadline
String	cardCode	Document No.
String	archivesCode	Archive No.
String	record	<Record>
String	issuingAuthority	Issuing unit
List< Integer >	recognizeWarnCode	List of recognition warning codes
List< String >	recognizeWarnMsg	Description of recognition warning codes

VehicleLicenseCardResult

Result entity class of driving license recognition

Type	Name	Meaning
VehicleFrontInfo	vehicleFrontInfo	Home page information of driving license
VehicleBackInfo	vehicleBackInfo	Sub-page information of driving license
List< Integer >	recognizeWarnCode	List and definitions of recognition Code alarm codes
List< String >	recognizeWarnMsg	Description of recognition warning codes

VehicleFrontInfo

Result entity class of home page recognition of driving license

Type	Name	Meaning
String	plateNo	License plate number
String	vehicleType	Vehicle Type
String	owner	Owner
String	address	Address
String	useCharacter	Usage nature
String	model	Brand and model
String	vin	Vehicle identification number
String	engineNo	Engine number
String	registerDate	Registration date
String	IssueDate	Issue date
String	seal	Seal

VehicleBackInfo

Result entity class of vice page recognition of driving license

Type	Name	Meaning
String	plateNo	License plate number
String	fileNo	Archive No.
String	allowNum	Approved number of persons
String	totalMass	Gross weight
String	curbWeight	Curb weight
String	loadQuality	Approved load weight
String	externalSize	Overall dimensions
String	marks	Notes
String	record	Inspection log

String	totalQuasiMass	Gross towing weight

HKIDCardOcrResult

Entity class for common recognition results of Hong Kong (China) identity card versions 03 and 18

Type	Name	Meaning
String	cnName	Chinese name
String	enName	English name
String	telexCode	Chinese name code
String	sex	Gender: "Male M" or "Female F"
String	birthday	Date of Birth
String	permanent	Permanent resident identity card. 0: Non-permanent 1: Permanent -1: Unknown.
String	idNum	ID Number
String	symbol	Document symbol, symbol under date of birth, such as "***AZ"
String	firstIssueDate	First issue date
String	currentIssueDate	Most recent collection date
String	fakeDetectResult	True or false judgment. 0: Unable to judge (image is blurry, incomplete, reflective, too dim, etc.); 1: False; 2: True. Note: This field may return null, indicating that no valid value was found.
List<Integer>	warningCode	Multiple alarm codes, returned when the identity card is a rephotograph, photocopy, or Photoshop image. -9102: License photocopy alarm -9103: License rephotograph alarm

		-9104: License Photoshop alarm -9105: License anti-counterfeit alarm

PermitOcrResult

Recognition result entity class for Hong Kong (China) and Macao (China) travel permit

Type	Name	Meaning
String	name	Name
String	englishName	English name
String	number	Document No.
String	sex	Gender
String	validDate	Validity period
String	issueAuthority	Issuing authority
String	issueAddress	Place of issue
String	birthday	Date of Birth

MLIDPassportOcrResult

Entity class for international passport recognition result

Type	Name	Meaning
String	id	Passport ID
String	name	Name
String	issuingCountry	Issuing country
String	nationality	Nationality
String	dateOfBirth	Date of Birth
String	sex	Gender (F for female, M for male)
String	dateOfExpiration	Validity Period
List<Integer>	warn	Alarm code -9103 License rephotograph alarm

		-9102 License copy alarm -9106 Document occlusion alarm
String	image	Document image

String	advancedInfo	Extension field:
		{
		ID:{
		Confidence:0.9999
		},
		Name:{
		Confidence:0.9996
		}
		}

String	codeSet	MRZ Code sequence on the first line at the bottom
String	codeCrc	MRZ Code sequence on the second line at the bottom

HmtResidentPermitOcrResult

Entity class for recognition result of residence permit for Hong Kong (China), Macao (China), and Taiwan (China)

Type	Name	Meaning
String	name	Document Name
String	sex	Gender
String	birth	Date of Birth
String	address	Address
String	idCardNo	ID No.
int	cardType	0-Front 1-Back
String	validDate	Document validity period
String	authority	Issuing authority
String	visaNum	Issuance frequency
String	passNo	Passport number

ISDKKitResultListener

Callback class for OCR recognition results, used to receive recognition results and error exceptions.

```
/**
 * Callback class for OCR recognition result
 */
public interface ISDKKitResultListener {

    /**
     * Successful OCR recognition result
     *
     * @param response Recognition result JSON information
     * @param ocrProcessResult OCR recognition result with additional
information
     */
    void onProcessSucceed(String response, OcrProcessResult
ocrProcessResult);

    /**
     * OCR recognition exception
     *
     * @param errorCode //Error code
     * @param message Exception information
     * @param ocrProcessResult OCR recognition result with additional
information
     */
    void onProcessFailed(String errorCode, String message,
OcrProcessResult ocrProcessResult);
}
```

OcrProcessResult contains additional content of the recognition result.

OcrProcessResult

Additional data content carried after OCR recognition.

Type	Name	Meaning	Default Value

String	requestId	The requestId corresponding to this identify request (null if the network request fails).	Null
String	imageBase64Str	The base64 of the image used for this recognition.	Null

Example of response result for identity card front request:

```
{
    "Name": "Li Ming"
    "Sex": "Male",
    "Nation": "Han",
    "Birth": "1987/1/1",
    "Address": "Tencent Building, High-tech Park, Shijingshan District,
Beijing",
    "IdNum": "440524******010014",
    "Authority": "",
    "ValidDate": "",
    "AdvancedInfo": "{}",
    "RequestId": "ab2c132e-9e1c-43d3-b0ef-9b4d80f00330"
 }
```

Example of response result for identity card back request:

```
{
    "Name": "",
    "Sex": "",
    "Nation": "",
    "Birth": "",
    "Address": "",
    "IdNum": "",
    "Authority": "Zhaoxian Public Security Bureau",
    "ValidDate": "2010.07.21-2020.07.21",
    "AdvancedInfo": "{}",
    "RequestId": "0d394478-6d4d-48fc-8b19-552415bf46de"
 }
```

Example of response result for bank card request:

```
{
    "BankInfo": "China Merchants Bank (03080000)",
    "ValidDate": "07/2023",
```

```
    "CardType": "Credit Card",
    "CardName": "China Merchants Bank Credit Card",
    "BorderCutImage": null,
    "CardNoImage": null,
    "WarningCode": [
      -9110
    ],
    "CardNo": "6225768888888888",
    "RequestId": "68f8fcbf-6004-0111a-ac18-6f1a9308fs100mab"
  }
```

Example of response result for business card request:

```
{
   "BusinessCardInfos": [
     {
       "Name": "name",
       ```json
     },
     {
       "Name": "position",
       "Value": "Visual Designer"
     },
     {
       "Name": "department",
       "Value": "Social Platform Department"
     },
     {
       "Name": "company"
       "Value": "Tencent Tencent"
     },
     {
       "Name": "address"
       "Value": "Tencent Building, Keji Zhongyi Road, High-tech Park,
Nanshan District, Shenzhen"
     },
     {
       "Name": "mailbox",
       "Value": "ab***fg@tencent.com"
     },
     {
       "Name": "mobile phone",
       "Value": "+86-133****5678"
```
```

```
      },
      {
        "Name": "QQ",
        "Value": "1234567"
      },
      {
        "Name": "WeChat",
        "Value": "amy001"
      }
    ],
    "RetImageBase64": "",
    "RequestId": "98f8fcbf-933a-4e95-ac48-6f1a9308fs6h"
}
```

Example of response result for vehicle VIN:

```
{
  "Response": {
    "Vin": "LBV2B25G2E5069977",
    "RequestId": "c59d9002-6c8c-426d-b57f-a8837dee2c7c"
  }
}
```

Example of response result for vehicle license plate:

```
{
  "Response": {
    ```json
    "Confidence": 99,
    "Rect": {
      "X": 217,
      "Y": 233,
      "Width": 170,
      "Height": 21
    },
    "RequestId": "210103d3-db06-4691-abe0-c0853aae606b"
  }
}
```

Example of response result for driving license recognition:

```
{
```
```

```
    "Name": "Li Ming"
    "Sex": "Male",
    "Nationality": "China"
    "Address": "6F, Tengyun Building, 397 Tianlin Road, Xuhui District,
Shanghai",
    "DateOfBirth": "1987-01-01",
    "IssuingAuthority": "Shanghai Traffic Police Brigade, Shanghai
Public Security Bureau"
    "DateOfFirstIssue": "2011-10-01",
    "Class": "C1",
    "StartDate": "2011-10-01",
    "EndDate": "2017-10-01",
    "CardCode": "440524198701010014",
    "ArchivesCode": "",
    "Record": "",
    "RecognizeWarnCode": [
      -9106
    ],
    "RecognizeWarnMsg": [
      "WARN_DRIVER_LICENSE_PS_CARD"
    ],
    "RequestId": "4ba2958b-e7cf-41c2-aafe-fdc985307f63"
  }
```


Example of response result for driving license main page recognition:


```
{
    "FrontInfo": {
      "PlateNo": "Hu AA1234",
      "VehicleType": "Compact Car",
      "Owner": "Li Ming",
      "Address": "6F, Tengyun Building, 397 Tianlin Road, Xuhui
District, Shanghai",
      "UseCharacter": "Non-commercial",
      "Model": "Buick SGM7151LAAA",
      "Vin": "ABCDEFGH123456789",
      "EngineNo": "8B54321",
      "RegisterDate": "2011-10-10",
      "IssueDate": "",
      "Seal": "Shanghai Traffic Police Brigade, Shanghai Public Security
Bureau"
    },
    "BackInfo": null,
```

```
    "RecognizeWarnCode": [
      -9106
    ],
    "RecognizeWarnMsg": [
      "WARN_DRIVER_LICENSE_PS_CARD"
    ],
    "RequestId": "820916b4-b391-40a8-9203-7ae87e3f1954"
  }
```


Example of response result for driving license secondary page recognition:


```
{
    "FrontInfo": null,
    "BackInfo": {
      "PlateNo": "Wan AC4L50",
      "FileNo": "A00-00476807",
      "AllowNum": "5 people",
      "TotalMass": "1837kg",
      "CurbWeight": "1431kg",
      "LoadQuality": "--",
      "ExternalSize": "4620×1780×1498mm",
      "TotalQuasiMass": "--",
      "Marks": "",
      "Record": "Inspection valid until November 2015 Wan A(4S)"
    },
    "RecognizeWarnCode": [
      -9106
    ],
    "RecognizeWarnMsg": [
      "WARN_DRIVER_LICENSE_PS_CARD"
    ],
    "RequestId": "e2ebfaa0-19d3-4d2b-9ed8-7c7c21eb2b74"
  }
```


Example of response result for residence permit recognition for Hong Kong (China), Macao (China), and Taiwan (China):


```
{
  "Response": {
    "Name": "Li Youyou",
    "Sex": "Female",
    "Birth": "1997/6/2",
```

```
    "Address": "No. 1528, Gumei Road, Xuhui District, Shanghai, Building
A1, Tencent YouTu No.1",
    "IdCardNo": "830000199706020042",
    "CardType": 0,
    "ValidDate": "",
    "Authority": "",
    "VisaNum": "0",
    "PassNo": "000",
    "RequestId": "f72e7048-f1e0-42f3-b0bf-f8730d48bb5c"
  }
}
```


Example of response result for Hong Kong (China) identity card recognition:


```
{
  "Response": {
    "CnName": "Shenneng",
    "EnName": "SAN, Nan",
    "TelexCode": "300000000000",
    "Sex": "Female",
    "Birthday": "01-01-2001",
    "Permanent": 1,
    "IdNum": "C000000(E)",
    "Symbol": "***AZ",
    "FirstIssueDate": "(09-99)",
    "CurrentIssueDate": "23-09-10",
    "FakeDetectResult": 1,
    "WarningCode": [
      -9102,
      -9103
    ],
    "HeadImage": "xx",
    "RequestId": "fba1c9ad-aeb3-4418-9ecf-80ab1b5fc875"
  }
}
```


Example of response result for travel permit recognition for Hong Kong (China), Macao (China), and Taiwan (China):


```
{
  "Response": {
    "Name": "Li Ming"
```

```
    "EnglishName": "LIMING",
    "Number": "C00000000",
    "Sex": "Male",
    "ValidDate": "2018.10.09-2028.10.08",
    "IssueAuthority": "Exit and Entry Administration of the Ministry of
Public Security",
    "IssueAddress": "Guangdong"
    "Birthday": "1981.08.03",
    "RequestId": "3090debe-3662-4ef1-8784-6ef2fb59f75e"
  }
}
```

Example of response result for international passport recognition:

```
{
  "Response": {
    "ID": "E6918C",
    "Name": "LIM HEG CHUN STEE",
    "IssuingCountry": "SGP",
    "Nationality": "SGP",
    "DateOfBirth": "",
    "Sex": "M",
    "DateOfExpiration": "230414",
    "Warn": [],
    "Image": "",
    "AdvancedInfo": "{\"IssuingCountry\":
{\"Confidence\":\"0.9500\"},\"Name\":{\"Confidence\":\"0.9500\"},\"ID\":
{\"Confidence\":\"0.9500\"},\"Nationality\":
{\"Confidence\":\"0.9500\"},\"Sex\":
{\"Confidence\":\"0.9500\"},\"DateOfExpiration\":
{\"Confidence\":\"0.9500\"}}",
    "CodeSet": "P<CANBROWN<<BENJAMIN<<<<<<<<<<<<<<<<<<<<<<<",
    "CodeCrc": "JK123488<3CAN8510168M2603298<<<<<<<<<<<<<<04",
    "RequestId": "0ee989d3-d064-45ec-bccb-63f5064247b4"
  }
}
```

For the returned error code and error information, please refer to Error Code.

ISdkOcrEntityResultListener

Callback class for OCR recognition results, used to receive recognition result entities and error exceptions.

```
public interface ISdkOcrEntityResultListener<T> {

    /**
     * Successful OCR recognition result
     *
     * @param t Returns entity information
     * @param ocrProcessResult OCR recognition result with additional
information
     */
    void onProcessSucceed(T t, OcrProcessResult ocrProcessResult);

    /**
     * OCR recognition exception
     *
     * @param errorCode //Error code
     * @param message Exception information
     * @param ocrProcessResult OCR recognition result with additional
information
     */
    void onProcessFailed(String errorCode, String message,
OcrProcessResult ocrProcessResult);
}
```

OcrProcessResult contains additional content of the recognition result.

Error Codes

Common error codes

Error Code	Description
UnsupportedOperation	Operation is not supported.
ResourceInUse	Resource is occupied.
InternalError	An internal error occurs.
RequestLimitExceeded	The number of requests exceeded the frequency limit.
AuthFailure.SecretIdNotFound	The key does not exist. Please check in the console whether the key has been deleted or disabled. If the status is normal, check whether the key is filled in correctly. Note that there should be no spaces before and after it.

LimitExceeded	Exceeded quota limit.
NoSuchVersion	The API version does not exist.
ResourceNotFound	The resource does not exist.
AuthFailure.SignatureFailure	Signature error. The signature calculation is incorrect. Please check the signature calculation process against the signature method documentation in the call method.
AuthFailure.SignatureExpire	Signature expired. The difference between the timestamp and server time must not exceed five minutes. Please check if the local time is synchronized with the standard time.
UnsupportedRegion	The API does not support the specified region.
UnauthorizedOperation	Unauthorized operation.
InvalidParameter	Parameter error.
ResourceUnavailable	Resources are unavailable.
AuthFailure.MFAFailure	MFA error.
AuthFailure.UnauthorizedOperation	Request unauthorized. Please refer to the CAM documentation for authentication details.
AuthFailure.InvalidSecretId	Invalid key (not TencentCloud API key type).
AuthFailure.TokenFailure	Token error.
DryRunOperation	DryRun operation indicates the request would be successful, but the DryRun parameter was included.
FailedOperation	Operation failed.
UnknownParameter	Unknown parameter error.
UnsupportedProtocol	HTTP(S) request protocol error, only GET and POST requests are supported.
InvalidParameterValue	Parameter value error.
InvalidAction	API does not exist.
MissingParameter	Parameter missing error.

ResourceInsufficient	Insufficient resources.

Business error codes

Error Code	Description
FailedOperation.ArrearsError	Account is in arrears.
FailedOperation.CountLimitError	Today's request count has reached the limit.
FailedOperation.DetectFailed	Detection failed.
FailedOperation.DownLoadError	File download failed.
FailedOperation.EmptyImageError	The image content is empty.
FailedOperation.EngineRecognizeTimeout	Engine recognition timeout.
FailedOperation.IdCardInfoIllegal	Identity card information is invalid (e.g., invalid identity card number or name field).
FailedOperation.ImageBlur	The image is blurry.
FailedOperation.ImageDecodeFailed	Image decoding failed.
FailedOperation.ImageNoBusinessCard	No business card detected in the photo.
FailedOperation.ImageNoIdCard	No identity card detected in the image.
FailedOperation.ImageNoText	No text detected in the image.
FailedOperation.ImageSizeTooLarge	Image size is too large. Please refer to the output parameters for the size limit.
FailedOperation.InvoiceMismatch	Invoice data inconsistency.

FailedOperation.LanguageNotSupport	The input language is not supported.
FailedOperation.MultiCardError	There are multiple cards in the photo.
FailedOperation.OcrFailed	OCR recognition failed.
FailedOperation.UnKnowError	Unknown error.
FailedOperation.UnOpenError	Service not activated.
InvalidParameter.ConfigFormatError	Config is not a valid JSON format.
InvalidParameter.EngineImageDecodeFailed	Image decoding failed.
InvalidParameter.InvalidGTINError	Invalid GTIN.
InvalidParameterValue.InvalidParameterValueLimit	Parameter value error.
LimitExceeded.TooLargeFileError	File content is too large.
ResourceNotFound.NoInvoice	Invoice does not exist.
ResourcesSoldOut.ChargeStatusException	Billing status is exceptional.

Local sdk error codes

Error Code	Description
OcrSdk.UserCancelOcr	User actively stopped OCR.
OcrSdk.InnerOcrError	OCR internal error, please try again.
OcrSdk.CallInitFirst	SDK needs to call init first.
FailedOperation.OcrFaile	Service recognition OCR result abnormal

d	
OcrSdk.InnerOcrErrorClipError	SDK internal image selection and crop logic failed
OcrSdk.PermissionError	Callback for user denied permission
OcrSdk.PermissionError_GoSetting	Callback for user denied permission and actively went to the Settings interface

Harmony API Overview

Last updated: 2025-02-06 16:16:54

The Harmony OCR SDK mainly involves the following classes: API interface class OcrSDKKit, parameter configuration class OcrSDKConfig, result callback class OcrResultCallback.

OcrSDKKit

OcrSDKKit is the external API class of the OCR SDK, and the main logic is also completed by calling this class.

API	Feature Description
init()	Initialization API
release()	Resource release API
startOcrByConfig()	Start OCR recognition
setOcrEventListener()	Set the event callback for this OCR
stopOcr()	Disable OCR
getVersion()	Get the current SDK version number

Init()

```
public init(context: Context)
```

Feature introduction: OCR SDK initialization API.
Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
Context	context	App context information

Release()

```
public release()
```

Feature introduction: OCR SDK resource release API.

StartOcrByConfig()

```
public startOcrByConfig(config: OcrSDKConfig, cb: OcrResultCallback)
```

Feature introduction: OCR SDK startup API.
Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
OcrSDKConfig	config	SDK startup configuration items
OcrResultCallback	cb	Result callback class

SetOcrEventListener()

```
public setOcrEventListener(callback: OcrEventCallback)
```

Feature introduction: Set the event callback for this OCR recognition
Input parameters:

Parameter Type	Parameter Name	Parameter Meaning
OcrEventCallback	callback	Callback information for the event

StopOcr()

```
public stopOcr()
```

Feature introduction: Stop OCR recognition.

GetVersion()

```
public getVersion(): string
```

Feature introduction: Get the current version number of the OCR SDK

OcrSDKConfig

OcrSDKConfig is the configuration entity class when starting the OCR SDK, mainly including the following properties

Type	Name	Meaning	Default Value
OcrMode	ocrMode	Card recognition mode	OCR_DETECT_AUTO_MANUAL
number	auto2ManualTimeout	Automatic capture timeout (in milliseconds, minimum 5 seconds, internal limit 30 seconds)	20000 ms
OcrCardType	cardType	Card recognition type	COMMON_OCR
OcrUIConfig	ocrUIConfig	Customize UI configurations	Default Value
string	secretID	Key information used for the request (if using fixed key mode, a fixed key can be passed in)	Null
string	secretKey	Key information used for the request (if using fixed key mode, a fixed key can be passed in)	Null
string	token	Temporary token information used for the request	Null
boolean	cropIdCard	Enable ID card photo cropping (remove extra edges outside the document, automatically correct shooting angle) switch	false
boolean	cropPortrait	Portrait photo crop (automatically extract the ID card portrait area)	false
boolean	copyWarn	Enable copy alarm	false
boolean	borderCheckWarn	Enable border and in-frame occlusion alarm	false
boolean	reshootWarn	Enable re-photographing alarm	false
boolean	detectPsWarn	Enable PS detection alarm	false
boolean	tempIdW	Enable temporary identity card alarm	false

	arn		
boolean	invalidDateWarn	Enable invalid identity card validity date alarm	false
boolean	quality	Enable image quality score (evaluate image blurriness)	false
boolean	multiCardDetect	Whether to enable front and back recognition in the same frame (only supports front and back recognition of the second-generation ID card or temporary identity card)	false
boolean	reflectWarn	Whether to enable reflection detection	false
boolean	retImageType	Image preprocessing, detect the skew angle of the image, rotate the originally skewed image around the center point to correct it, and finally output a corrected business card cutout.	false
boolean	retBorderCutImage	For bank cards, whether to return the preprocessed (precisely cropped and aligned) bank card image data	false
boolean	retCardNoImage	For bank cards, whether to return the cropped image data of the card number, default is false.	false
boolean	enableQualityValue	For bank cards, whether to return the image quality score (the image quality score is a standard for evaluating the blurriness of an image)	false

OcrCardType

Supported card recognition types for OCR

<OcrCardType Type>	Meaning
IDCardOCR_FRONT	ID card portrait side recognition mode
IDCardOCR_BACK	ID card national emblem side recognition mode

BankCardOCR	Bank card front recognition mode
BusinessCardOCR	Business card front recognition mode
LicensePlateOCR	Vehicle license plate recognition mode
VinOCR	Vehicle VIN recognition mode
VehicleLicenseOCR_FRONT	Driving license home page recognition mode
VehicleLicenseOCR_BACK	Driving license secondary page recognition mode
DriverLicenseOCR_FRONT	Driving license home page recognition mode
DriverLicenseOCR_BACK	Driving license secondary page recognition mode
IDCardOCR_HK03	Hong Kong (China) identity card version 03 recognition mode
IDCardOCR_HK18	Hong Kong (China) identity card version 18 recognition mode
Exit_Entry_HK_Macao_Card	Hong Kong (China), Macao (China), and Taiwan (China) travel permit recognition mode
HMT_RESIDENT_PERMIT_OCR	Residence permit for Hong Kong (China), Macao (China), and Taiwan (China)
MLID_PASSPORT	International passport recognition mode

OcrMode

OcrMode is an enumeration type that lists card recognition modes.

<OcrModeType Type>	Representation Of Meaning
OCR_DETECT_MANUAL	Manual shooting mode
OCR_DETECT_AUTO_MANUAL	Automatic identification mode (Note: After 20s, prompt whether to switch to manual shoot)
OCR_DETECT_AUTO	Automatic identification mode

OcrResultCallback

Callback class for OCR results

```
export interface OcrResultCallback {
  /**
   * Successful callback
   *
   * @param response Server-side returned information
   * @param requestID Server-side requestID
   * @param imageBase64Str Base64 encoding of the optimal image in jpg
format from the SDK
   */
  onSuccess: (response: string, requestID: string, imageBase64Str:
string) => void;
  /**
   * Failure callback
   *
   * @param errorCode error code
   * @param errMsg detailed error information
   */
  onFail: (errorCode: number, errMsg: string) => void;
}
```

OcrEventCallback

Used to listen to callbacks for events during the OCR process

```
export interface OcrEventCallback {
  /**
   * Additional parameter passthrough
   * @param type event type
   * @param params
   */
  onOcrEvent: (type: OcrEventType, param: string) => void;
}
```

OcrEventType

Corresponding event types and meanings:

Event Enumeration	Event Meaning
EVENT_DETECT_RECT_INFO	Detection frame information (coordinates of the top-left corner of the detection frame and

	the width and height of the detection frame, params in the form: "{ x: 0, y: 0, w: 0, h: 0}")
EVENT_TIME_OUT	Timeout event callback (countdown start, suspend, end, params in the form: start, pause, end)
EVENT_AUTO_TO_MANUAL	Event callback when switching from automatic to manual (params are meaningless, can be ignored)

OcrUIConfig

Parameter configuration for custom UI of OCR detection page

Type	Name	Meaning	Default Value
boolean	showLightIcon	Whether to display the flash button	true
boolean	showAlbumIcon	Whether to display the album selection button	true
boolean	showFixTips	Whether to display the 'Please avoid folded corners...' note	true
boolean	showRectCornerLine	Whether to display the lines at the four corners	true
boolean	showIDCardLogo	Whether to display the head and National Emblem contours (Mainland identity card)	true
boolean	keepDetectView	Whether to terminate the compatibility check page after detecting the best frame (requires custom animation effects)	false
number	lightIconSize	Width of the flash icon, height is adaptive	50
number	takePictureIconSize	Width of the photo icon, height is adaptive	80
number	albumIconSize	Width of the album icon, height is adaptive	50

string	detectNormalColor	Color of general feedback tips	#ffffff
string	detectCorrectColor	Color of normal feedback tips	#006eff
string	detectErrorColor	Color of exceptional feedback tips	#e94b2c
number	rectCornerLineWeight	Thickness of the detection frame corner lines	3
number	rectCornerLineLen	Length of the detection frame corner lines	30
string	maskViewColorStyle	Mask color style of the detection page	rgba(80, 80, 80, 0.8)
number	rectMarginTop	Distance from the detection frame to the top	50
number	rectHorizontalMargin	Horizontal distance from the detection frame to the screen	5
number	tipsMarginTop	Distance from the tips Note to the top	20
number	fixTipsMarginTop	Fix the distance from the tips Note to the top	350
string	switchModeDialogTitleColor	Color of the timeout dialog title	#000000
string	switchModeDialogContentColor	Color of the timeout dialog content	#000000
string	switchModeDialogCancelColor	Color of the timeout dialog cancel button	#000000
string	switchModeDialogConfirmColor	Color of the timeout dialog confirmation button	#006eff
number	switchModeDialogTitleF	Font size of the timeout dialog title	16

	ontSize		
number	switchModeDialogContentFontSize	Font size of the timeout dialog content	14
number	switchModeDialogCancelFontSize	Font size of the timeout dialog cancel button	16
number	switchModeDialogConfirmFontSize	Font size of the timeout dialog confirmation button	16
boolean	switchModeDialogConfirmOnLeft	Whether the timeout dialog confirmation button is displayed on the left	false
(param: CommonUIModel) => void	commonCustomUI	Basic custom UI (see custom UI documentation for details)	(param: CommonUIModel): void => {}
(param: AnimationUIModel) => void	animUI	Custom animation UI (see custom UI documentation for details)	(param: AnimationUIModel): void => {}

Custom UI

Custom UI Display

OCR SDK supports Custom UI, which can be configured through OcrUIConfig. Usage is as follows:

```
// 1. Custom configuration
let uiConfig = new OcrUIConfig();
// Whether to display the flash button on the recognition page
uiConfig.showLightIcon = this.showLightIcon;
// Whether to display the album button on the recognition page
uiConfig.showAlbumIcon = this.showAlbumIcon;
// Detect if the 'Please avoid folding the content...' note is displayed
on the page
uiConfig.showFixTips = this.showFixTips;

// 2. Set uiConfig to startup parameters
```

```
let config: OcrSDKConfig = new OcrSDKConfigBuilder()
  .setOcrUIConfig(uiConfig)
  .build();

// 3. Pass in startup parameters
OcrSDKKit.getInstance().startOcrByConfig(config, ...);
```

Adding a UI Component

This section mainly introduces how to add custom UI components on the Identity verification page and how to bind events. Usage is as follows:

```
/**
 * Custom UI
 * @param $$
 */
@Builder
MyCustomUI($$: CustomUIModel) {
  Column() {
    Button('Custom Button').onClick(() => {
      // Custom UI event response
      router.back();
    }).margin({
      bottom: '3%'
    }).backgroundColor(Color.Blue)
  }.backgroundColor(Color.Red)
  .margin({
    bottom: '3%'
  })
}

// 1. Custom configuration
let uiConfig = new OcrUIConfig();
// Pass in custom layout
uiConfig.commonCustomUI = this.MyCustomUI;

// 2. Set uiConfig to startup parameters
let config: OcrSDKConfig = new OcrSDKConfigBuilder()
  .setOcrUIConfig(uiConfig)
  .build();
```

```
// 3. Pass in startup parameters
OcrSDKKit.getInstance().startOcrByConfig(config, ...);
```

Custom Prompt

If you need to customize the prompt text on the OCR page, you can achieve it as follows (the OCR SDK delivery includes a file to be translated named ocr_custom_string.json, which contains all customizable text):

1. Open the module named <module_name> (the module integrated with the OCR SDK).
2. Add `ocr_custom_string.json` to the corresponding language folder (<module_name>/src/main/resources/zh_CN/element/).
3. Modify the text content in `ocr_custom_string.json` that needs customization.
4. After packaging, the content you modified will overwrite the original prompt text of the SDK.

Personal Information Protection Policy For Optical Character Recognition (OCR) Client SDK (New Version)

Last updated: 2025-02-06 16:17:04

You can view the OCR client SDK Personal Information Protection Rules by clicking the following link:
Click to view >>