

# PalmSDK-L User Guide

V1.3.8-2024/09/09

## Directory

- 1 SDK Introduction..... 6**
  - 1.1 Overview..... 6**
  - 1.2 Directory structure..... 6**
  - 1.3 Applicable system.....6**
- 2 SDK Usage process.....7**
  - 2.1 Configuration description.....7**
    - 2.1.1 gradle Configure..... 7
    - 2.1.2 Androidmanifest.xml Configure..... 7
    - 2.1.3 Flow chart.....9
- 3 SDK Interface description.....10**
  - 3.1 Description of stageScores..... 10**
  - 3.2 Class file description..... 12**
  - 3.3 PalmSdk.....14**
    - 3.3.1 Initialize..... 14
    - 3.3.2 Get SDK version number..... 14
  - 3.4 Device..... 14**
    - 3.4.1 Create device..... 14
    - 3.4.2 Close device..... 14
    - 3.4.3 Obtain the number of devices..... 15
  - 3.5 Device DeviceListener..... 16**
  - 3.6 DtUsbManager DeviceStateListener..... 16**
  - 3.7 IDevice..... 16**
    - 3.7.1 Open device..... 16

3.7.2 Obtain the stream types supported by the device .....	17
3.7.3 Create Stream .....	17
3.7.4 Destroy Stream .....	17
3.7.5 Close device .....	17
3.7.6 Reboot device .....	17
<b>3.8 IOpenCallback .....</b>	<b>19</b>
<b>3.9 IStream .....</b>	<b>19</b>
3.9.1 Get stream type .....	19
3.9.2 Apply for frame list space .....	19
3.9.3 Get frame list .....	19
3.9.4 Start stream .....	20
3.9.5 Stop stream .....	20
<b>3.10 IVeinshine .....</b>	<b>20</b>
3.10.1 Get device information .....	20
3.10.2 Set Psensor threshold .....	20
3.10.3 Set Led mode .....	21
3.10.4 Upgrade .....	21
3.10.5 Start heartbeat .....	21
3.10.6 Stop heartbeat .....	22
3.10.7 Obtain camera module temperature .....	22
3.10.8 Enable dimpalm algorithm module .....	22
3.10.9 Read license .....	22
3.10.10 Write license .....	23
3.10.11 Get algorithm version .....	23
3.10.12 Capture once .....	23
3.10.13 Continuous capture .....	24
3.10.14 Stop capture .....	25
3.10.15 Obtain recognition threshold .....	25
3.10.16 Start saving image .....	25
3.10.17 Stop saving images .....	26
3.10.18 Registration Interface .....	27
3.10.19 Extract feature from image .....	27
3.10.20 Create PalmClient .....	27
3.10.21 Register to server .....	29

3.10.22 Delete featureId .....	29
3.10.23 Query featureId from server .....	29
3.10.24 Get license from server .....	30
<b>3.11 Frames .....</b>	<b>31</b>
3.11.1 Get the total number of frames .....	31
3.11.2 Get frames .....	31
<b>3.12 Frame .....</b>	<b>31</b>
3.12.1 Get stream type .....	31
3.12.2 Get frame type .....	31
3.12.3 Get frame width .....	31
3.12.4 Get frame height .....	32
3.12.5 Get frame index .....	32
3.12.6 Get frame size .....	32
3.12.7 Get frame timestamp .....	32
3.12.8 Obtain raw data .....	32
3.12.9 Get extra frame information .....	32
<b>3.13 BBox class .....</b>	<b>33</b>
<b>3.14 DeviceInfo class .....</b>	<b>33</b>
<b>3.15 ICapturePalmCallback .....</b>	<b>33</b>
3.15.1 Capture palm callback .....	33
3.15.2 No palm callback captured .....	33
<b>3.16 IHeartbeatListener .....</b>	<b>35</b>
3.16.1 Heartbeat callback .....	35
<b>3.17 IUpgradeListener .....</b>	<b>35</b>
3.17.1 Upgrade start callback .....	35
3.17.2 Upgrade progress callback .....	35
3.17.3 Upgrade successful callback .....	35
3.17.4 Upgrade failure callback .....	35
3.17.5 Upgrade timeout callback .....	35
<b>3.18 CameraTemperature .....</b>	<b>37</b>
<b>3.19 HeartbeatParam .....</b>	<b>37</b>

3.20 CaptureFrame .....	38
3.21 FrameType enumeration .....	40
3.22 Hint .....	40
3.23 StreamType .....	46
3.24 EnumRecognitionType .....	46
3.25 PalmRegisterOutput .....	47
3.26 ExtraFrameInfo .....	48
3.27 ClientPalmOutput .....	48
3.28 ImageInstance .....	48
3.29 ExtractOutput .....	48
<b>4 Palm Recognition and Registration Process – Threshold Configuration .....</b>	<b>50</b>
4.1 Threshold Configuration for Liveness and Quality in the Palm Recognition Process .....	50
4.2 Threshold Configuration for Liveness and Quality in the Registration Process .....	51
<b>5 Mobile phone registration .....</b>	<b>52</b>
5.1 Overview .....	52
<b>5.2 PhoneCameraPalmManager .....</b>	<b>52</b>
5.2.1 Open device .....	52
5.2.2 Enable dimpalm algorithm module .....	52
5.2.3 Get algorithm version .....	52
5.2.4 Create PalmClient .....	53
5.2.5 Extract feature from image .....	53
5.2.6 Image feature value comparison .....	53
5.2.7 Register To Server .....	54
5.2.8 Query featureId from server .....	54
5.2.9 Delete featureId .....	55
5.2.10 Device close .....	55
<b>6 Error codes and descriptions .....</b>	<b>55</b>
6.1 Universal error code .....	55

6.2 PalmClient and server interaction error codes.....	59
7 <i>Revision Record</i> .....	61
8 <i>Disclaimer</i> .....	61
9 <i>Technical support</i> .....	61
10 <i>Precautions</i> .....	62

# 1 SDK Introduction

## 1.1 Overview

The PalmSDK is a software development toolkit developed based on the Veinshine camera, which is currently suitable for the Android platform and provides a series of friendly APIs and simple application example programs for application developers. Users can obtain high-precision color and grayscale images based on this development package, which facilitates the development of applications such as biometric recognition and artificial intelligence perception. This document introduces the usage of PalmSDK (Android) on the Android platform. Including usage process, interface explanation, and example program usage.

## 1.2 Directory structure

The following table shows the directory structure and content description of PalmSDK

Directory	Description
doc	Description document directory
example	Source code directory for sample programs
example-apk	APK file directory for demo
libs	Lib file directory
assets	Facial Algorithm Model File Directory

## 1.3 Applicable system

Type	Environment
arm64-8a	Android 6.0 and above
armabi-v7a	Android 6.0 and above

## 2 SDK Usage process

### 2.1 Configuration description

#### 2.1.1 gradle Configure

Copy the file to the corresponding libs directory of the project, and then configure it in the gradle.

reference is as follows:

```
android{
    sourceSets {
        main {
            jniLibs.srcDirs = ["libs"]
        }
    }
    repositories {
        flatDir {
            dirs 'libs'
        }
    }
}

dependencies {
    implementation files('palm-android-sdk-v1.3.7.jar')
}
```

#### 2.1.2 Androidmanifest.xml Configure

Add permissions to the manifest file

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

<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"
    tools:ignore="ProtectedPermissions" />
<uses-permission android:name="android.hardware.usb.host" />
<uses-permission android:name="android.hardware.usb.accessory" />

<uses-feature android:name="android.hardware.camera" />
```

```
<uses-feature  
  android:name="android.hardware.usb.host"  
  android:required="true" />
```



### 2.1.3 Flow chart

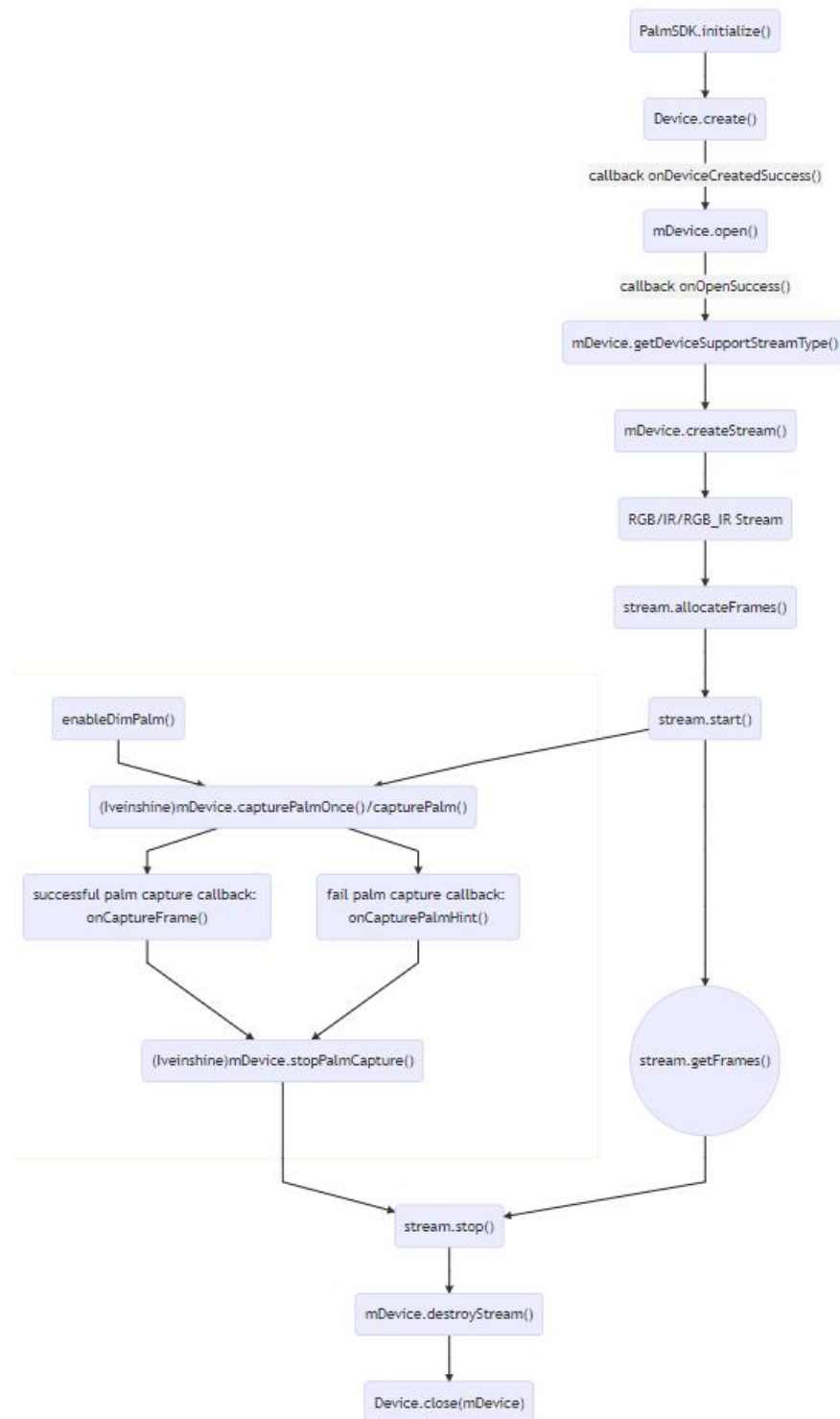


chart 1

## 3 SDK Interface description

### 3.1 Description of stageScores

stageScores are of Map<Integer,Float> type, and the int value of their keys corresponds to the value value of the Hint class.

Different interfaces return different scores, and the key-value pairs they contain change with the return value of the interface.

The detailed relationship between the returned results and the keys contained in stageScores is shown in the table below

Interface name	description	result	A list of keys contained in Scores
onCapturePalmHint of the ICapturePalmCallback class (Hint hint, HashMap<Integer, Float> stageScores)	The first argument, hint, indicates that the interface returns a result.  (Corresponding to the third column of the table returned by the interface)  The key value in the second parameter, stageScores, corresponds to the key of Scores (the list of keys in the fourth column of the table, Scores).	PALM_IR_QUALITY_ERROR(2)	PALM_IR_QUALITY_ERROR(2);
		PALM_IR_LIVENESS_ERROR(4)	PALM_IR_QUALITY_ERROR(2); PALM_IR_LIVENESS_ERROR(4)
		PALM_RGB_LIVENESS_COLOR_GRAY_ERROR(14)	PALM_IR_QUALITY_ERROR(2); PALM_IR_LIVENESS_ERROR(4); PALM_RGB_LIVENESS_COLOR_GRAY_ERROR(14);
		PALM_RGB_RELIABILITY_ERROR(16)	PALM_BIG_POSE_YAW(1); PALM_RGB_RELIABILITY_ERROR(16);
		PALM_IR_RELIABILITY_ERROR(15)	PALM_BIG_POSE_YAW(1); PALM_RGB_RELIABILITY_ERROR(16); PALM_IR_RELIABILITY_ERROR(15);
The onCaptureFrame interface of the ICapturePalmCallback class	Callback The default result of this interface is	PALM_SUCCESS(0)	PALM_BIG_POSE_YAW(1); PALM_RGB_RELIABILITY_ERROR(16); PALM_IR_RELIABILITY_ERROR(15);

	PALM_SUCCES S,  The second parameter is the same as stageScores		
extractPalmFeatu resFromImg	The result in the ExtractOutp ut class returns a result for the interface.  Same as above for stageScores	PALM_IR_QUALITY_ERROR(2)	PALM_IR_QUALITY_ERROR(2);
		PALM_IR_LIVENESS_ERROR(4)	PALM_IR_QUALITY_ERROR(2); PALM_IR_LIVENESS_ERROR(4);
		PALM_SUCCESS(0)	PALM_IR_QUALITY_ERROR(2); PALM_IR_LIVENESS_ERROR(4); PALM_BIG_POSE_YAW(1);
registerPalm	PalmRegiste rOutput  result corresponds to the result returned by the interface.  Same as above for stageScores	PALM_IR_QUALITY_ERROR(2)	PALM_IR_QUALITY_ERROR(2);
		PALM_IR_REGISTER_QUALITY_ERR OR(3)	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERRO R(3);
		PALM_IR_LIVENESS_ERROR(4)	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERRO R(3); PALM_IR_LIVENESS_ERROR(4);
		PALM_RGB_LIVENESS_COLOR_GRAY _ERROR(14)	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERRO R(3); PALM_IR_LIVENESS_ERROR(4); PALM_RGB_LIVENESS_COLOR_GRAY _ERROR(14);
		PALM_RGB_PHONE_REGISTER_QUAL ITY_ERROR(29)	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERRO R(3); PALM_IR_LIVENESS_ERROR(4); PALM_RGB_LIVENESS_COLOR_GRAY _ERROR(14); PALM_RGB_PHONE_REGISTER_QUALI TY_ERROR(29);
		PALM_IR_RELIABILITY_ERROR(15 )	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERRO R(3); PALM_IR_LIVENESS_ERROR(4); PALM_RGB_LIVENESS_COLOR_GRAY _ERROR(14); PALM_RGB_PHONE_REGISTER_QUALI TY_ERROR(29); PALM_BIG_POSE_YAW(1); PALM_IR_RELIABILITY_ERROR(15) ;
		PALM_IR_REGISTER_RELIABILITY _ERROR(17)	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERRO R(3); PALM_IR_LIVENESS_ERROR(4); PALM_RGB_LIVENESS_COLOR_GRAY _ERROR(14); PALM_RGB_PHONE_REGISTER_QUALI TY_ERROR(29);

			PALM_BIG_POSE_YAW(1); PALM_IR_RELIABILITY_ERROR(15); PALM_IR_REGISTER_RELIABILITY_ERROR(17);
		PALM_RGB_RELIABILITY_ERROR(16)	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERROR(3); PALM_IR_LIVENESS_ERROR(4); PALM_RGB_LIVENESS_COLOR_GRAY_ERROR(14); PALM_RGB_PHONE_REGISTER_QUALITY_ERROR(29); PALM_BIG_POSE_YAW(1); PALM_IR_RELIABILITY_ERROR(15); PALM_IR_REGISTER_RELIABILITY_ERROR(17); PALM_RGB_RELIABILITY_ERROR(16);
		PALM_RGB_REGISTER_RELIABILITY_ERROR(18)	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERROR(3); PALM_IR_LIVENESS_ERROR(4); PALM_RGB_LIVENESS_COLOR_GRAY_ERROR(14); PALM_RGB_PHONE_REGISTER_QUALITY_ERROR(29); PALM_BIG_POSE_YAW(1); PALM_IR_RELIABILITY_ERROR(15); PALM_IR_REGISTER_RELIABILITY_ERROR(17); PALM_RGB_RELIABILITY_ERROR(16); PALM_RGB_REGISTER_RELIABILITY_ERROR(18);
		PALM_RGB_PHONE_REGISTER_RELIABILITY_ERROR(30)	PALM_IR_QUALITY_ERROR(2); PALM_IR_REGISTER_QUALITY_ERROR(3); PALM_IR_LIVENESS_ERROR(4); PALM_RGB_LIVENESS_COLOR_GRAY_ERROR(14); PALM_RGB_PHONE_REGISTER_QUALITY_ERROR(29); PALM_BIG_POSE_YAW(1); PALM_IR_RELIABILITY_ERROR(15); PALM_IR_REGISTER_RELIABILITY_ERROR(17); PALM_RGB_RELIABILITY_ERROR(16); PALM_RGB_REGISTER_RELIABILITY_ERROR(18); PALM_RGB_PHONE_REGISTER_RELIABILITY_ERROR(30);

## 3.2 Class file description

Class name	Content Description
PalmSdk	Environmental management class

Device	Equipment management class
Device DeviceListener	Device creation callback interface class
DtUsbManager DeviceStateListener	Device status callback interface class
IDevice	Device Interface Class
IVeinshine	Specific device interface classes
IOpenCallback	Device Open Callback Interface Class
IStream	Stream Interface Class
Frames	Frame List Class
Frame	Frame data class
DeviceInfo	Camera information object class
FrameMode	Frame mode enumeration class
FrameType	Frame type enumeration class
StreamType	Stream type enumeration class
CaptureFrame	Capture callback frame object class
Hint	Capture prompts enumeration class
CameraTemperature	Camera temperature object
HeartbeatParam	Heartbeat parameter object
EnumRecognitionType	Identifying type enumeration classes
ICapturePalmCallback	Capture callback interface class
IUpgradeListener	Upgrade callback interface class
IHeartbeat Listener	Heartbeat callback interface class
PalmRegisterOutput	Register palm interface result output class
ExtraFrameInfo	Information class when registering interface input parameters
ClientPalmOutput	PalmClient output class
ImageInstance	Image entity class
ExtractOutput	Extract features from images output class

## 3.3 PalmSdk

### 3.3.1 Initialize

```
static void initialize()
```

Initialization method, please initialize once when the app starts

Parameters:

### 3.3.2 Get SDK version number

```
static String getSdkVersion ()
```

Parameters:

Return SDK version number

## 3.4 Device

### 3.4.1 Create device

```
static void create(@NonNull Context context,@NonNull DeviceListener  
deviceCreateListener,DtUsbManager.DeviceStateListener deviceStateListener)
```

Parameters:

[in]	context	Context of Activity
[in]	deviceCreatListener	Device creation result callback object, see 3.4<DeviceListener>interface description
[in]	deviceStateListener	Device status callback object, see 3.5<DeviceStateListener>interface description

### 3.4.2 Close device

```
void close(IDevice device)
```

Parameters:

[in]	device	Device that need to be close
------	--------	------------------------------

### 3.4.3 Obtain the number of devices

```
int getDeviceCount(Context context)
```

Parameters:

[in]	context	Context
------	---------	---------

**Return** the number of inserted devices

### 3.5 Device DeviceListener

Device creation callback interface	
void on DeviceCreated Success ( IDevice device, Int deviceIndex, Map<Long, IDevice>runningDevice, UsbMapTable DeviceType	Device creation success callback interface
void on DeviceCreate Failed (IDevice device)	Device creation failure callback interface
void on DeviceDestroy (IDevice device)	Device destruction callback interface

### 3.6 DtUsbManager DeviceStateListener

Device status callback interface	
void onDevicePermissionGranted (UsbDevice usbDevice)	Device USB Permission Granted Callback Interface
void onDevicePermissionDenied (UsbDevice usbDevice)	Device USB permission Denied callback interface
void on Attached (UsbDevice usbDevice)	Device Attached Callback Interface
void on Detached (UsbDevice usbDevice)	Device Detached callback interface

### 3.7 IDevice

#### 3.7.1 Open device

int open (IOpenCallback openCallback)	
Parameters:	
[in]	openCallback      Open callback, see 3.7<IOpenCallback>
Return    success: 0	
failure: see error code	



### 3.7.2 Obtain the stream types supported by the device

List<StreamType>getDeviceSupportStreamType()

Parameters:

**Return** success: StreamType supported by the device, see 3.22<StreamType>  
failure: see error code

### 3.7.3 Create Stream

IStream createStream (StreamType streamType)

Parameters:

[in]	streamType	Stream type
------	------------	-------------

**Return** success: Stream object, see 3.8<IStream>  
failure: Empty object

### 3.7.4 Destroy Stream

void destroyStream (IStream stream)

Parameters:

[in]	stream	Stream Object
------	--------	---------------

### 3.7.5 Close device

int close()

Parameters:

**Return** success: 0  
failure: see error code

### 3.7.6 Reboot device

int reboot()

Parameters:

Return success: 0

failure: see error code

## 3.8 IOpenCallback

Open callback interface	
void onDownloadPrepare()	Download preparation callback interface
void onDownloadProgress (int progress)	Download progress callback interface
void onDownloadSuccess()	Download successful callback interface
void onOpenSuccess()	Open successful callback interface
void onOpenFailure (int errorCode)	Open failed callback interface

## 3.9 IStream

### 3.9.1 Get stream type

StreamType getStreamType()	
Parameters:	
Return	success: Stream type, see 3.22<StreamType> failure: Empty object

### 3.9.2 Apply for frame list space

Frames allocateFrames()	
Parameters:	
Return	success: Frame object, see 3.10<Frames> failure: Empty object

### 3.9.3 Get frame list

int getFrames (Frames frames, int timeout)	
Parameters:	
[in]	frames      Get frames
[in]	timeout      timeout time

**Return** success: 0  
failure: see error code

### 3.9.4 Start stream

**int start()**

**Parameters:**

**Return** success: 0  
failure: see error code

### 3.9.5 Stop stream

**void stop()**

**Parameters:**

## 3.10 IVeinshine

### 3.10.1 Get device information

**DeviceInfo getDeviceInfo()**

**Parameters:**

**Return** success: DeviceInfo object, see 3.13<DeviceInfo>  
Failure: Empty object

### 3.10.2 Set Psensor threshold

**int setPsensorDistanceThreshold (int nearThreshold, int farThreshold)**

**Parameters:**

[in]	nearThreshold	Close range threshold
[in]	farThreshold	Long distance threshold

**Return** success: 0  
failure: see error code

### 3.10.3 Set Led mode

int setLedMode (int mode)

Parameters:

[in]	mode	Light mode
------	------	------------

**Return** success: 0  
failure: see error code

### 3.10.4 Upgrade

int upgrade(@NonNull String upgradeFilePath, @NonNull IUpgradeListener listener)

Parameters:

[in]	upgradeFilePath	Upgrade file path
[in]	listener	Upgrade callback listening, see 3.16<IUpgradeListener>

**Return** success: 0  
failure: see error code

### 3.10.5 Start heartbeat

int startHeartbeat(@NonNull HeartbeatParam heartbeatParam, @NonNull IHeartbeatListener listener)

Parameters:

[in]	heartbeatParam	Heartbeat parameters, see 3.18<HeartbeatParam>
[in]	listener	Heartbeat callback monitoring, see 3.15<IHeartbeatListener>

**Return** success: 0  
failure: see error code

### 3.10.6 Stop heartbeat

int stopHeartbeat()

Parameters:

Return success: 0  
failure: see error code

### 3.10.7 Obtain camera module temperature

int getCameraTemperature(@NonNull CameraTemperature temperature)

Parameters:

[in]	temperature	Camera temperature object, see 3.17<CameraTemperature>
------	-------------	---

Return success: 0  
failure: see error code

### 3.10.8 Enable dimpalm algorithm module

int enableDimPalm(String modelPath)

Parameters:

[in]	modelPath	Model path
------	-----------	------------

Return success: 0  
failure: see error code

### 3.10.9 Read license

String readLicense()

Parameters:

Return success:String license  
failure:Empty String

### 3.10.10 Write license

int writeLicense(String licenseContext)

Parameters:

[in]	licenseContext	License context
------	----------------	-----------------

Return    success:0  
          failure:see error code

### 3.10.11 Get algorithm version

String getAlgorithmVersion()

Parameters:

Return    success:Algorithm version  
          failure:Empty String

### 3.10.12 Capture once

int capturePalmOnce(@NonNull ICapturePalmCallback capturePalmCallback, int timeout,boolean isGetBase64)

Parameters:

[in]	capturePalmCallback	Capture callback monitoring, see 3.14<ICapturePalmCallback>
[in]	timeout	Capture timeout
[in]	isGetBase64	Whether the snapshot callback obtains the Base64 encoded data

Return    success: 0  
          failure: see error code

### 3.10.13 Continuous capture

```
int capturePalm(@NonNull ICapturePalmCallback capturePalmCallback, int  
timeout,boolean isGetBase64)
```

#### Parameters:

[in]	capturePalmCallback	Capture callback monitoring, see 3.14<ICapturePalmCallback>
[in]	timeout	Capture timeout
[in]	isGetBase64	Whether the snapshot callback obtains the Base64 encoded data

**Return**    success: 0  
             failure: see error code



### 3.10.14 Stop capture

int stopPalmCapture()

Parameters:

Return success: 0

failure: see error code

### 3.10.15 Obtain recognition threshold

float getRecognitionThreshold(@NonNull EnumRecognitionType type)

Parameters:

[in]	type	Identification type, see 3.23<EnumRecognitionType>
------	------	---

Return success: 0

failure: see error code

### 3.10.16 Start saving image

public int startSavePicture(@NonNull String picPath, int saveCount, int minFreeSpaceMb)

public int startSavePicture(@NonNull String picPath, int saveCount, int minFreeSpaceMb, ISaveFailedCallback saveFailedCallback)

Parameters:

[in]	picPath	Save image path
[in]	saveCount	Number of stored images
[in]	minFreeSpaceMb	Minimum free disk space (in MB) to maintain. Saving will stop if free space $\leq$ this value.
[in]	saveFailedCallback	This callback function is mainly used to prompt file storage failure when there is insufficient space.

Return success: 0

failure: see error code

### 3.10.17 Stop saving images

int stopSavePicture()

Parameters:

Return success: 0

failure: see error code

### 3.10.18 Registration Interface

PalmRegisterOutput registerPalm(ImageInstance rgbImg, @NonNull ImageInstance irlmg, ExtraFrameInfo info)

#### Parameters:

[in]	rgbImg	RGB image ,see 3.27<ImageInstance>
[in]	irlmg	IR image ,see 3.27<ImageInstance>
[in]	info	Extra frame information, see 3.28<ExtractOutput>

**Return** success: PalmRegisterOutput.result==0  
failure: Empty object or result != 0

### 3.10.19 Extract feature from image

ExtractOutput extractPalmFeaturesFromImg(ImageInstance rgbImg, @NonNull ImageInstance irlmg)

#### Parameters:

[in]	rgbImg	RGB image ,see 3.27<ImageInstance>
[in]	irlmg	IR image ,see 3.27<ImageInstance>

**Return** success: ExtractOutput.result==0, see 3.28<ExtractOutput>  
failure: Empty object or result != 0

### 3.10.20 Create PalmClient

boolean createPalmClient(@NonNull String companyId, @NonNull String sn, @NonNull String ip, @NonNull String port, String hostName);

#### Parameters:

[in]	companyId	Company ID
[in]	sn	SN
[in]	ip	IP address

[in]	port	port
[in]	hostName	Host name
<b>Return</b>	success:true	
	failure:false	

### 3.10.21 Register to server

```
ClientPalmOutput registerToServer(ImageInstance rgbImg, byte[] rgbFeature, @NonNull ImageInstance irlImg, @NonNull byte[] irFeature);
```

#### Parameters:

[in]	rgbImg	RGB Image, see 3.27<ImageInstance>
[in]	rgbFeature	RGB Image feature
[in]	irlImg	IR Image, see 3.27<ImageInstance>
[in]	irFeature	IR Image feature

**Return** success:ClientPalmOutput != null && ClientPalmOutput.resultCode==0  
failure:ClientPalmOutput.resultCode != 0

### 3.10.22 Delete featureId

```
int deleteId(int featureId)
```

#### Parameters:

[in]	featureId	featureId
------	-----------	-----------

**Return** success:0  
failure:See error code

### 3.10.23 Query featureId from server

```
ClientPalmOutput queryFeatureIdFromServer(ImageInstance rgbImg, byte[] rgbFeature, @NonNull ImageInstance irlImg, @NonNull byte[] irFeature)
```

#### Parameters:

[in]	rgbImg	RGB Image, see 3.27<ImageInstance>
[in]	rgbFeature	RGB Image feature
[in]	irlImg	IR Image, see 3.27<ImageInstance>
[in]	irFeature	IR Image feature

**Return** success:ClientPalmOutput != null && ClientPalmOutput.resultCode==0  
failure:ClientPalmOutput.resultCode != 0

### 3.10.24 Get license from server

String getLicenseFromServer()

Parameters:

**Return** success:String license  
failure:Empty String

## 3.11Frames

### 3.11.1Get the total number of frames

int getFrameCount()

Return the number of frames

### 3.11.2Get frames

Frame getFrame (int frameIndex)

Parameters:

[in]	frameIndex	Frame index
------	------------	-------------

Return success: Frame object, see 3.11<Frame>  
failure: null

## 3.12Frame

### 3.12.1Get stream type

StreamType getStreamType()

Return the StreamType of the frame, see 3.22<StreamType>

### 3.12.2Get frame type

FrameType getFrameType()

Return the FrameType of the frame, see 3.20<FrameType>

### 3.12.3Get frame width

int getWidth()

Return int type frame width

### 3.12.4Get frame height

`int getHeight()`

Return    frame height of type int

### 3.12.5Get frame index

`int getIndex()`

Return    int frame Index

### 3.12.6Get frame size

`int getSize()`

Return    int frame size

### 3.12.7Get frame timestamp

`long getTimestamp()`

Return    long type timestamp

### 3.12.8Obtain raw data

`byte[] getRawData()`

Return    byte[]

### 3.12.9Get extra frame information

`ExtraFrameInfo getExtraInfo()`

Return    ExtraFrameInfo, see 3.28<ExtraFrameInfo>



### 3.13 BBox class

BBox class attribute description	
int x	Top left vertex coordinates
int y	Top left vertex coordinates
int w	width
int h	height

### 3.14 DeviceInfo class

Description of camera version information attributes	
String devicename	Device Name
String serialnum	SN number
String palm_sdk_version	Palm SDK version number
String firmware_version	Firmware Version
int pid	PID
int vid	Vid

### 3.15 ICapturePalmCallback

#### 3.15.1 Capture palm callback

void onCaptureFrame (CaptureFrame frame)	
Parameters:	
[in] frame	Capture successfully callback data frames, see <CaptureFrame>, Description of stageScores

#### 3.15.2 No palm callback captured

void onCapturePalmHint(Hint hint, HashMap<Integer, Float> stageScores)
--

Parameters:

[in]	hint	Callback prompt for failed capture, see <Hint>
[in]	stageScores	Palm stage scores. For specific details, please refer to the stageScores instructions

## 3.16 IHeartbeatListener

### 3.16.1Heartbeat callback

void onHeartbeatResult (boolean result)

Parameters:

[in]	result	Heartbeat callback
------	--------	--------------------

## 3.17 IUpgradeListener

### 3.17.1Upgrade start callback

void onUpgradeStart()

### 3.17.2Upgrade progress callback

void onUpgradeProgress (int progress)

Parameters:

[in]	progress	Upgrade progress callback
------	----------	---------------------------

### 3.17.3Upgrade successful callback

void onUpgradeSuccess()

### 3.17.4Upgrade failure callback

void onUpgradeFailure (String msg)

Parameters:

[in]	msg	Failed callback information
------	-----	-----------------------------

### 3.17.5Upgrade timeout callback

void onUpgradetimeout()



### 3.18 CameraTemperature

Description of camera temperature attributes	
float temperatureMainBoard	Main board temperature
float temperatureLedBoard	Lamp panel temperature
float temperatureCpu	CPU temperature
float temperatureRgbSensor	RGB Sensor temperature

### 3.19 HeartbeatParam

Description of heartbeat parameter properties	
int heartbeatInterval	heartbeat interval
int FailTimes	Number of failed retries

### 3.20CaptureFrame

Description of callback properties for successful capture results	
int rgbCols	Number of RGB image columns
int rgbRows	RGB image row count
byte[] rgbData	RGB image data (8UC3)
String rgbBase64	RGB image coding data (Base64)
int irCols	Number of IR image columns
int irRows	IR Image Rows
byte[] irData	IR image data (8UC1)
String irBase64	IR image coding data (Base64)
int palmRectX	The x-coordinate of the upper left vertex of the palm frame
int palmRectY	The y-coordinate of the upper left vertex of the palm frame
int palmRectW	The width of the palm frame
int palmRectH	The height of the palm frame
int palmCenterRectX	The x-coordinate of the upper left vertex of the palm center box
int palmCenterRectY	The y-coordinate of the upper left vertex of the palm center box
int palmCenterRectW	The width of the palm center frame
int palmCenterRectH	The height of the palm center frame
byte[] rgbFeature	Palm RGB feature
byte[] irFeature	Palmar IR feature
byte[] skeleton	Palm Key Points Group
int[] pSensorValue	Psensor value
Map<Integer, Float> stageScores	Palm stage scores, see the stageScores instructions for details

int palmType	Palm type (0: Left hand 1: Right hand)
--------------	--

### 3.21 FrameType enumeration

Description of FrameType enumeration values	
INVALID_FRAME_TYPE	Illegal frame type
RGB_FRAME	RGB frame type
IR_FRAME	IR frame type

### 3.22 Hint

Capture callback prompt enumeration class	
int key	Code for indicating prompt language
String value	Description of prompt language
String chineseDescription	Description of prompt Chinese language

Enum	Key	description	chineseDescription
NO_PALM_DETECTED	-1	No palm detected	未检测到手掌
PALM_SUCCESS	0	Brush palm successfully	刷掌成功
PALM_BIG_POSE_YAW	1	Too big pose	角度过大
PALM_IR_QUALITY_ERROR	2	Please face your palm towards the camera	请将掌心面向镜头
PALM_IR_REGISTER_QUALITY_ERROR	3	Please open and straighten	请张开并摆正手掌



		your palms	
PALM_IR_LIVENESS_ERROR	4	Please face your palm towards the camera	请将掌心面向镜头
PALM_IR_DARKNESS	6	Please palm slightly closer	手掌稍微靠近点
PALM_IR_OVER_EXPOSE	7	Please keep your palms slightly away	手掌稍微远离点
PALM_RGB_DARKNESS	8	Please palm slightly closer	手掌稍微靠近点
PALM_RGB_OVER_EXPOSE	9	Please keep your palms slightly away	手掌稍微远离点
PALM_IR_REGISTER_DARKNESS	10	Please palm slightly closer	手掌稍微靠近点
PALM_IR_REGISTER_OVER_EXPOSE	11	Please keep your	手掌稍微远离点

		palms slightly away	
PALM_RGB_REGISTER_DARKNESS	12	Please palm slightly closer	手掌稍微靠近点
PALM_RGB_REGISTER_OVER_EXPOSE	13	Please keep your palms slightly away	手掌稍微远离点
PALM_RGB_LIVENESS_COLOR_GRAY_ERROR	14	Please face your palm towards the camera	请将掌心面向镜头
PALM_IR_RELIABILITY_ERROR	15	Please ensure that your palms are clear and free from any abnormalities	请确保手掌清晰 无异常
PALM_RGB_RELIABILITY_ERROR	16	Please ensure that your palms are clear and free	请确保手掌清晰 无异常

		from any abnormalities	
PALM_IR_REGISTER_RELIABILITY_ERROR	17	Please ensure that your palms are clear and free from any abnormalities	请确保手掌清晰无异常
PALM_RGB_REGISTER_RELIABILITY_ERROR	18	Please ensure that your palms are clear and free from any abnormalities	请确保手掌清晰无异常
PALM_IR_AE_DARKNESS	19	The Ir image in the firmware input ROI is too dark	固件传入 ROI 内 Ir 图像过暗
PALM_IR_AE_OVER_EXPOSE	20	Ir overexposure in firmware	固件传入 ROI 内 Ir 过曝

		input ROI	
PALM_RGB_AE_DARKNESS	21	The Rgb image in the firmware input ROI is too dark	固件传入 ROI 内 Rgb 图像过暗
PALM_RGB_AE_OVER_EXPOSE	22	Rgb overexposure in firmware input ROI	固件传入 ROI 内 Rgb 过曝
PALM_UNEXPECTED_CENTER_BOX_POS	23	Please place your palm in the center of the screen	手掌请位于画面中心
PALM_ILLEGAL_ENV	24		
PALM_IS_MOVING	25	Please keep your palm still	手掌请保持静止
PALM_INVALID_LIGHT_STATUS	26		
PALM_RGB_QUALITY_ERROR	27	Please face your palm towards the camera	请张开并摆正手掌
PALM_DEFAULT	28		
PALM_RESULT_PHONE_RGB_REGISTER_QUALITY_ERROR	29	The RGB registration	手机 RGB 注册质量不合格

		quality of the mobile phone is not qualified	
PALM_RESULT_REGISTER_RELIABILITY_PHONE_RGB_ERROR	30	The RGB registration reliability of the mobile phone is not qualified	手机 RGB 注册可靠性不合格
MANUALSTOP	0x4000	Manually stopped	手动停止
TIMEOUT	0x8000	Capture timeout	抓拍超时
INITIALIZING	0x8888	Initializing	初始化中

### 3.23 StreamType

StreamType enumeration value description	
INVALID_STREAM_TYPE	Illegal stream type
RGB	RGB stream type
IR	IR stream type
RGB_IR	RGB and IR synchronous stream types

### 3.24 EnumRecognitionType

Description of the enumeration values for the EnumRecognitionType	
RGB	RGB recognition type
IR	IR recognition type

### 3.25 PalmRegisterOutput

Palm registration interface result output class	
int result	Result
Map<Integer, Float> stageScores	Palm stage scores, see the stageScores instructions for details
byte[] rgbFeature	Palm RGB feature
byte[] irFeature	Palm IR feature
byte[] skeleton	Palm Key Points Group
int palmType	Palm type (0: Left hand 1: Right hand)
int palmRectX	The x-coordinate of the upper left vertex of the palm frame
int palmRectY	The y-coordinate of the upper left vertex of the palm frame
int palmRectW	The width of the palm frame
int palmRectH	The height of the palm frame
int palmCenterRectX	The x-coordinate of the upper left vertex of the palm center box
int palmCenterRectY	The y-coordinate of the upper left vertex of the palm center box
int palmCenterRectW	The width of the palm center frame
int palmCenterRectH	The height of the palm center frame

### 3.26 ExtraFrameInfo

Extra frame Information class	
int[] pSensorValue	Psensor value
int[] palmRoi	Palm ROI
int lightMode	Light mode

### 3.27 ClientPalmOutput

PalmClient Output class	
int resultCode	Error code output by PalmClient
String resultMsg	Output information prompt by PalmClient
int featureId	The featureId output by PalmClient

### 3.28 ImageInstance

ImageInstance class	
int width	The width of the image
int height	The Height of the image
byte[] imgData	The data of the image
ImageFormat format	The format of the image(IMG_1C8BIT:8UC1 IMG_3C8BIT:8UC3)

### 3.29 ExtractOutput

Extract feature from image result output class	
int result	Result
Map<Integer, Float> stageScores	Palm stage scores, see the stageScores instructions for details
byte[] rgbFeature	Palm RGB feature
byte[] irFeature	Palm IR feature



byte[] skeleton	Palm Key Points Group
int palmType	Palm type (0: Left hand 1: Right hand)

# 4 Palm Recognition and Registration Process

## – Threshold Configuration

### 4.1 Threshold Configuration for Liveness and Quality in the Palm Recognition Process

Quality Threshold Configuration in `dim_status_pipeline_config.pbtxt`:

The palm quality score ranges between 0.0 and 1.0. The configuration below means that when the palm quality score falls between 0.0 and 0.5, it is considered invalid, and the pipeline executes `seq=999` as an error handling procedure and terminates. Otherwise, it proceeds to the operator with `seq=3`.

You can modify the `high_limit` to adjust the pass rate of palm quality. A lower `high_limit` value increases the pass rate, but it also increases the false positive rate for lower-quality palms.

```
operator_info{
  seq:2
  operator_type:kPalmQualityOperator
  decision_branches{
    next_seq:999
    key_parameters{
      param_name:"QualityScore"
      value_type:TypeScope
      scope_value{
        low_limit:0.0
        high_limit:0.5
      }
    }
  }
}
decision_branches{
  next_seq:3
}
```

Liveness Threshold Configuration in `dim_status_pipeline_config.pbtxt`:

The palm liveness score ranges between 0.0 and 1.0. The configuration below means that when the palm liveness score falls between 0.0 and 0.5, it is considered invalid, and the pipeline executes `seq=999` as an error handling procedure and terminates. Otherwise, it proceeds to the operator with `seq=3`.

You can modify the `high_limit` to adjust the pass rate for palm liveness detection. A lower `high_limit` value increases the pass rate, but it also increases the false positive rate for fake palms.

```
operator_info{
  seq:3
  operator_type:kPalmLivenessOperator
  decision_branches{
    next_seq:999
    key_parameters{
      param_name:"LivenessScore"
      value_type:TypeScope
    }
  }
}
```

```

        scope_value{
            low_limit:0.0
            high_limit:0.5
        }
    }
}
decision_branches{
    next_seq:3
}
}

```

## 4.2 Threshold Configuration for Liveness and Quality in the Registration Process

Quality Threshold Configuration in `dim_register_pipeline_config.pbtxt`:

The palm quality score ranges between 0.0 and 1.0. The configuration below means that when the palm quality score falls between 0.0 and 0.5, it is considered invalid, and the pipeline executes `seq=999` as an error handling procedure and terminates. Otherwise, it proceeds to the operator with `seq=3`. You can modify the `high_limit` to adjust the pass rate of palm quality. A lower `high_limit` value increases the pass rate, but it also increases the false positive rate for lower-quality palms.

```

operator_info{
    seq:2
    operator_type:kPalmQualityOperator
    decision_branches{
        next_seq:999
        key_parameters{
            param_name:"QualityScore"
            value_type:TypeScope
            scope_value{
                low_limit:0.0
                high_limit:0.5
            }
        }
    }
}
decision_branches{
    next_seq:3
}
}

```

Liveness Threshold Configuration in `dim_register_pipeline_config.pbtxt`:

The palm liveness score ranges between 0.0 and 1.0. The configuration below means that when the palm liveness score falls between 0.0 and 0.5, it is considered invalid, and the pipeline executes `seq=999` as an error handling procedure and terminates. Otherwise, it proceeds to the operator with `seq=3`. You can modify the `high_limit` to adjust the pass rate for palm liveness detection. A lower `high_limit` value increases the pass rate, but it also increases the false positive rate for fake palms.

```

operator_info{
    seq:3
    operator_type:kPalmLivenessOperator
    decision_branches{

```

```

        next_seq:999
        key_parameters{
            param_name:"LivenessScore"
            value_type:TypeScope
            scope_value{
                low_limit:0.0
                high_limit:0.5
            }
        }
    }
    decision_branches{
        next_seq:3
    }
}

```

## 5 Mobile phone registration

### 5.1 Overview

For mobile only (separate version)

### 5.2 PhoneCameraPalmManager

#### 5.2.1 Open device

PhoneCameraPalmManager Create();

Return success:PhoneCameraPalmManager  
fail:printout error

#### 5.2.2 Enable dimpalm algorithm module

int phoneEnablePalm(String modelPath)

参数:

[in]	modelPath	Modle path

Return success:0

fail:See error code

#### 5.2.3 Get algorithm version

String getPhonePalmAlgorithmVersion()

Return success:algorithm version

fail:null

## 5.2.4 Create PalmClient

```
boolean createPhonePalmClient(@NonNull String companyId, @NonNull String sn,  
@NonNull String ip, @NonNull String port, String hostName);
```

参数:

[String]	companyId	Company ID
[String]	sn	SN
[String]	ip	IP address
[String]	port	port
[String]	hostName	Host name

return success:true

fail:false

## 5.2.5 Extract feature from image

```
ExtractOutput extractPhonePalmFeaturesFromImg(ImageInstance rgbImg, @NonNull  
ImageInstance irlmg)
```

Parameters:

[in]	rgbImg	RGB image ,see 3.27<ImageInstance>
[in]	irlmg	IR image ,see 3.27<ImageInstance>

Return success:ExtractOutput.result==0, see 3.28<ExtractOutput>

failure:Empty object or result != 0

## 5.2.6 Image feature value comparison

```
CompareOutput compareFeatureScore(byte[] rgbFeatureSrc, byte[] irFeatureSrc, byte[]  
rgbFeatureDest, byte[] irFeatureDest)
```

参数:

[in]	rgblmg	RGB Image, see 3.27<ImageInstance>
[in]	rgbFeature	RGB Image feature
[in]	irlmg	IR Image, see 3.27<ImageInstance>
[in]	irFeature	IR Image feature

**Return** success:ClientPalmOutput != null && ClientPalmOutput.resultCode==0  
failure:ClientPalmOutput.resultCode != 0

### 5.2.7 Register To Server

ClientPalmOutput phonePalmRegisterToServer(ImageInstance rgblmg, byte[] rgbFeature, @NonNull ImageInstance irlmg, @NonNull byte[] irFeature);

Parameters:

[in]	rgblmg	RGB Image, see 3.27<ImageInstance>
[in]	rgbFeature	RGB Image feature
[in]	irlmg	IR Image, see 3.27<ImageInstance>
[in]	irFeature	IR Image feature

**Return** success:ClientPalmOutput != null && ClientPalmOutput.resultCode==0  
failure:ClientPalmOutput.resultCode != 0

### 5.2.8 Query featureId from server

ClientPalmOutput queryFeatureIdFromServer(ImageInstance rgblmg, byte[] rgbFeature, @NonNull ImageInstance irlmg, @NonNull byte[] irFeature)

Parameters:

[in]	rgblmg	RGB Image, see 3.27<ImageInstance>
[in]	rgbFeature	RGB Image feature
[in]	irlmg	IR Image, see 3.27<ImageInstance>

[in]	irFeature	IR Image feature
------	-----------	------------------

---

**Return**    success:ClientPalmOutput != null && ClientPalmOutput.resultCode==0  
               failure:ClientPalmOutput.resultCode != 0

### 5.2.9 Delete featureId

int phonePalmDeleteId(int featureId)
--------------------------------------

---

**return**    success:0  
               fail:See error code

### 5.2.10 Device close

void phonePalmClose(IDevice device)
-------------------------------------

---

**Parameters:**

[in]	device	Need to close
------	--------	---------------

## 6 Error codes and descriptions

### 6.1 Universal error code

Error (Hexadecimal)	Description
0x1	Unknown error
0x2	Unrealized
0x3	Invalid parameter
0x4	Not currently supported
0x5	Memory application failed
0x6	Invalid picture type
0x20010	Transfer failed
0x20012	The configuration file does not exist

0x21001	Unable to find device
0x21002	Null pointer
0x21003	Open failed
0x21004	Closing failed
0x21005	Open flow failure
0x21006	Failed to set/retrieve data
0x21007	Failed to detect data
0x21008	Failed to open IR camera
0x21009	Failed to open RGB camera
0x2100A	Failed to obtain USB serial number
0x2100B	The device is not running
0x2100C	Device not turned on
0x2100D	Driver error
0x2100E	Camera not configured
0x2100F	Stop stream fail
0x22001	Data size error
0x22002	Data not prepared
0x22004	Unsupported camera mode
0x22010	timeout
0x22011	Scan mode not set
0x22100	file does not exist
0x22101	Operation file failed
0x22102	Matching RGB data failed
0x22103	The upgraded version number has not changed
0x22104	The device is currently being upgraded
0x22105	<b>Upgrade fail</b>
0x22106	Set expose fail
0x22107	Get expose fail
0x23001	Failed to initialize facial algorithm



0x23002	Failed to initialize deep algorithm
0x23003	Invalid calibration size
0x23004	Failed to read flash
0x23005	Failed to obtain calibration
0x23006	Path error
0x23007	An error occurred
0x23008	Failed to obtain license
0x23009	Failed to init palm algorithm
0x24001	Camera component not found
0x24002	Failed to obtain RGB frames
0x24003	Failed to obtain IR frame
0x24004	Capturing in progress
0x24005	Preview open failed
0x24006	Preview read failed
0x25001	Algorithm not initialized
0x25002	Stream not start
0x25003	Device has opened
0x25004	Device not initialized
0x25005	Device is not in capture mode
0x25006	Invalid frame format
0x42019	SDK not configured successfully
0x42021	Unactivated algorithm
0x42023	Unopened flow
0x42024	Capture not enabled
0x80010	Operator creation failed
0x80011	Operator configuration file does not exist
0x80012	Model inference engine creation failed
0x80013	Algorithm pipeline creation failed
0x80014	The algorithm pipeline configuration file does not exist
0x80015	The pre - and post-processing module of the model does not exist

0x80016	Invalid running parameters
0x80017	Invalid configuration parameters
0x80020	No palm frame detected
0x80021	Failed to obtain the loop graph
0x80022	Failed to obtain norm graph
0x80024	Model inference failed
0x80030	Unexpected error occurred
0x80032	Null pointer occurred
0x80033	Light status error

## 6.2 PalmClient and server interaction error codes

Error (Decimal)	Description
1	Undefined error
2	Not implemented
3	Invalid argument
4	Not supported
5	Allocate memory fail
6	Deprecated
7	Exception occurred
10001	Invalid Http(s) Response
10002	Invalid Http(s) Status Code
10003	Deserialize Message Fail
10004	Query Score not pass
10005	Features is null or invalid format
10006	Connect error
10007	Handshake error
10008	Data send error
10009	Route not found
20001	URL invalid
20002	URL decodeFail fail
20003	Unregistered UID
20004	Unserialize fail
20005	Company id of UID mismatched
20006	Uid is invalid or expired
20007	Fail to decode RGB/IR features data
20008	Features hash version is invalid
30001	Connect database fail
30002	Insert fail
30003	Delete fail
30004	Query fail
30005	Query rgb and ir result is different
30006	Update Fail
30007	Register name duplicated
30008	Get last insert ID fail
30009	Company ID mismatched
30010	Features ID Not Found
30011	Try to register RGB features which has exits
30012	Try to register IR features which has exits

30013	Cannot registered without main features
30014	RGB features not registered
30015	IR features not registered

## 7 Revision Record

Version	Description	Date	Author
V1.0.1	SDK 1.0.1 version description document	2024/03/18	
V1.0.2	SDK 1.0.2 version description document	2024/03/22	
V1.1.0	SDK 1.1.0 version description document	2024/03/26	
V1.1.1	SDK 1.1.1 version description document	2024/03/28	
V1.1.2	SDK 1.1.2 version description document	2024/04/03	
V1.1.5	SDK 1.1.5 version description document	2024/04/16	
V1.1.6	SDK 1.1.6 version description document	2024/04/23	
V1.2.0	SDK 1.2.0 version description document	2024/05/10	
V1.2.2	SDK 1.2.2 version description document	2024/05/29	
V1.3.0	SDK 1.3.0 version description document	2024/07/11	
V1.3.3	SDK 1.3.3 version description document	2024/07/25	
V1.3.7	SDK 1.3.7 version description document	2024/08/29	
V1.3.8	SDK 1.3.8 version description document	2024/09/09	

## 8 Disclaimer

The device application information and other similar content described in this publication are for your convenience only and may be replaced by updated information. It is your own responsibility to ensure that the application complies with technical specifications.

## 9 Technical support

You can obtain support through the following channels

- FAE support: Please contact our sales personnel for FAE support

## 10 Precautions

- Do not use other heat sources to heat this product.
- Do not drop or impact this product to prevent damage to internal components and a decrease in accuracy; Improper operation may cause damage to internal components.
- Do not attempt to modify or disassemble this machine in any way to avoid module damage and decreased accuracy.
- After using the module for a period of time, it will generate heat, which is a normal phenomenon. Heat dissipation treatment can be performed on the back of the module.