# iScanPro development instructions document

Current Version: v1.3.2

Version Status: Released

Documentation Update Notes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Version | Update time | Adapted version | Updates | Participants |
| v1.0 | 2020/07/23 | V1.0 | Initial content | Wang Hang |
| v1.1 | 2020/11/12 | V1.1 | Update Interface | Lin Zhenyu / Dong Wenbin / Lu Quanfeng / Wang Hang |
| V1.2 | 2020/12/29 | V1.2 | Update interface definition | Lu Quanfeng/Wang Hang |
| V1.3.1 | 2022/1/6 | V1.3.1 | New Interface | Fan Zhi |
| V1.3.2 | 2022/7/18 | V1.3.2 | New optimized interface | Ouyang Qian |

**Catalog**

**[iScanPro development instructions document 1](#_Toc1026)**

**[1、 Document Description 4](#_Toc9301)**

**[2、 Development Notes 4](#_Toc17621)**

**[2.1. Configure the development environment 4](#_Toc14751)**

**[2.2, instantiate the interface, register the listener object 4](#_Toc2126)**

**[2.2, logging off the listening object 5](#_Toc17735)**

**[2.3, interface objects to call function functions 5](#_Toc17086)**

**[Description: See iScanProDemo example for details 5](#_Toc24294)**

**[3、Interface list 7](#_Toc30645)**

**[3.1、Open the scanning head 7](#_Toc9381)**

**[3.2. Close the scanning head 7](#_Toc14891)**

**[3.3. Start scanning 7](#_Toc23880)**

**[3.4. Stop scanning 8](#_Toc24282)**

**[3.5、Lock the scan button 8](#_Toc6038)**

**[3.6. Configure the output method of scanning results 8](#_Toc31354)**

**[3.7、Configuration of scanning beep, vibration, indicator 9](#_Toc6710)**

**[3.8. Additional terminator 10](#_Toc21178)**

**[3.9. Data processing rules 10](#_Toc20275)**

**[3.10, configure the trigger method 11](#_Toc24623)**

**[3.11, scan timeout time 12](#_Toc28857)**

**[3.12, continuous scanning interval 12](#_Toc12431)**

**[3.13. Set the character editing format 12](#_Toc10461)**

**[3.14、Whether to delete the existing content of the edit box 13](#_Toc13299)**

**[3.15. Restore default configuration parameters 13](#_Toc9313)**

**[3.16, decoding mode 14](#_Toc5365)**

**[3.17、Laser fill light mode 14](#_Toc22739)**

**[3.18、Configuration of barcode control switch 14](#_Toc31453)**

**[3.19. Picture output mode 15](#_Toc127)**

**[3.20、Multi-barcode enable 15](#_Toc20043)**

**[3.21、Configure multiple barcode precision mode 16](#_Toc8963)**

**[3.22、Configure the number of multiple barcodes 16](#_Toc7153)**

**[3.23. Registering scan result listener objects 16](#_Toc30611)**

**[3.24. Logout of the scan result listener object 17](#_Toc20060)**

**[3.25, callback interface class 17](#_Toc5665)**

**[4、Attachment page 19](#_Toc32486)**

**[4.1、Barcode type comparison table 19](#_Toc22280)**

**[Description: Code Type ID is int type 19](#_Toc24382)**

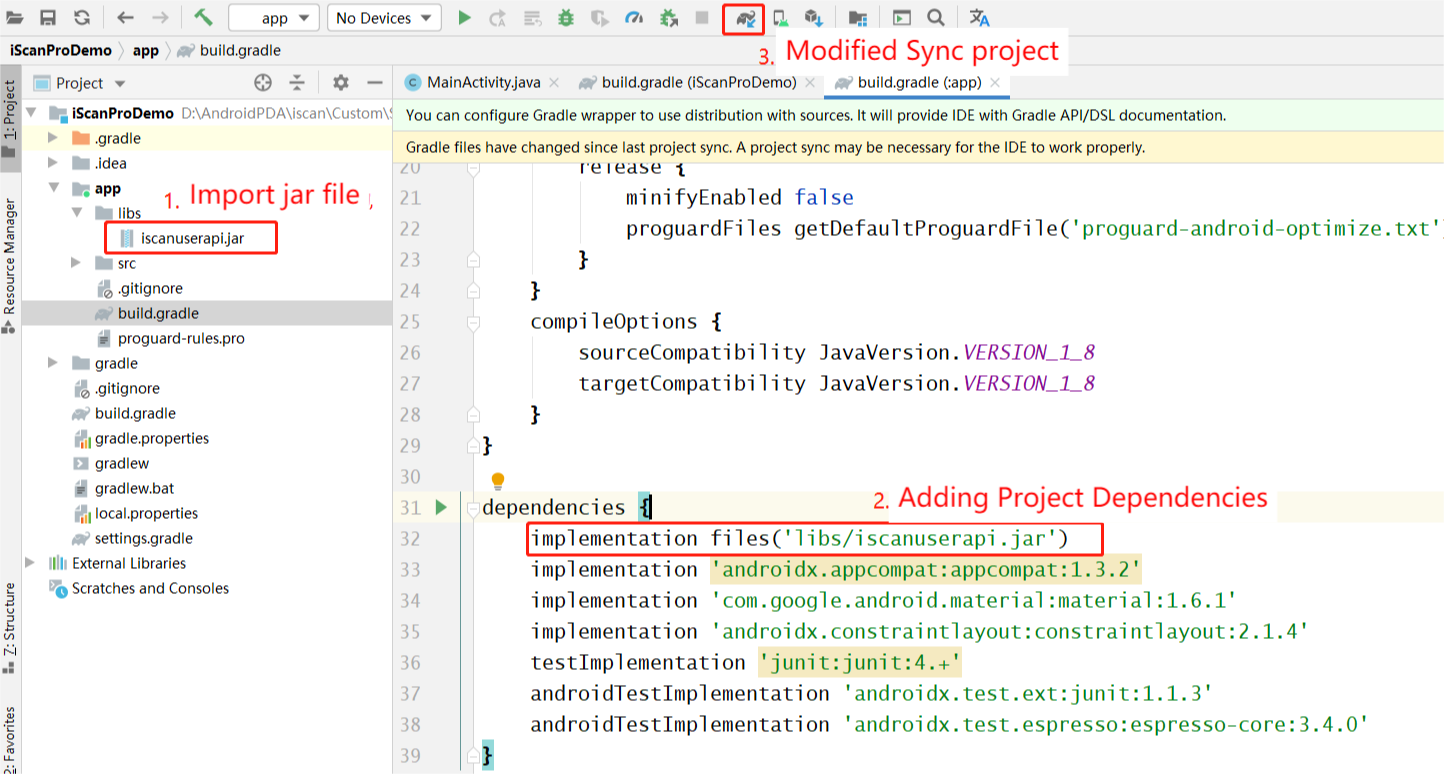
#### Document Description

This document mainly explains the iData terminal device sweeping development instructions for related industries and integrators off developers.

#### Development Notes

###### 2.1. Configure the development environment

Instructions：Create an AndroidStudio project or open an existing AndroidStudio project, configure the following environment

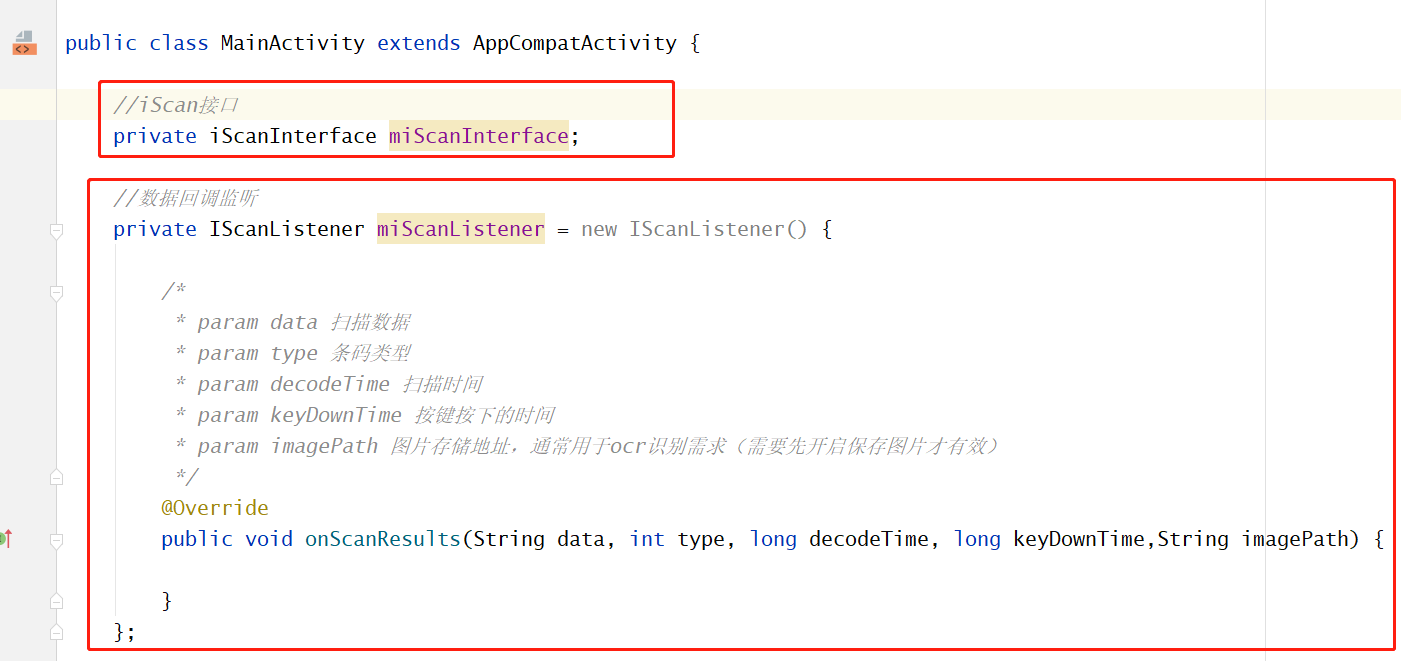


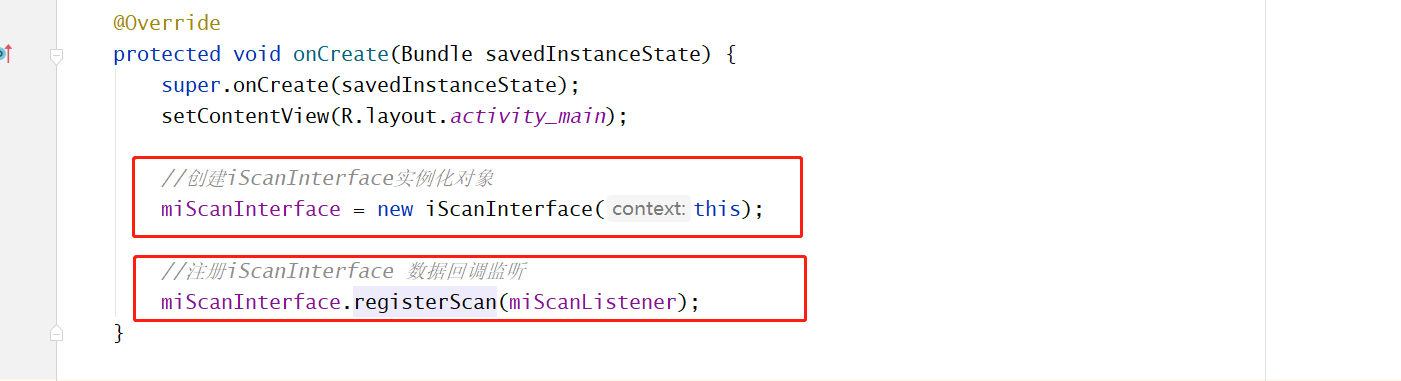
###### 2.2, instantiate the interface, register the listener object

**Description:** After registering the listener object, the object can be used to call the corresponding functional interface, to facilitate the development of third-party applications

**Note: iScan interface object and data listener object an APP can only use one, do not multiple Activity to register, if you need to do the global object, you can initialize the registration listener in the Application.**

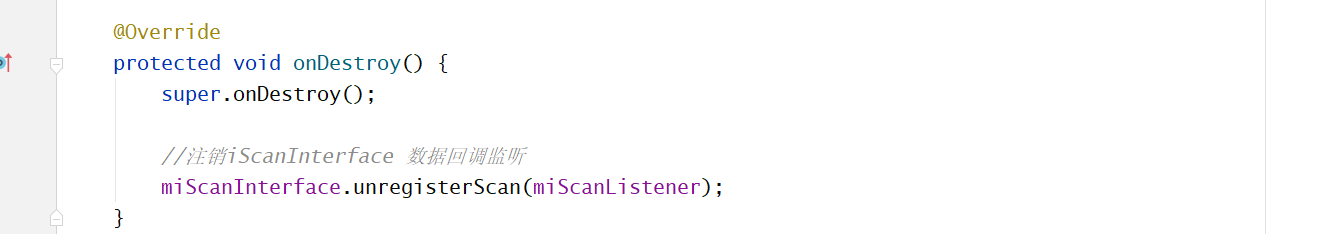
The following is an example of Activity registration:





###### 2.2, logging off the listening object

**Description:Log out of** listening when exiting the application



###### 2.3, interface objects to call function functions

**Description:** Calling a functional interface

**Description**: See iScanProDemo example for details



#### 

#### 3、Interface list

###### 3.1、Open the scanning head

|  |  |
| --- | --- |
| **void open( )** | |
| **Description** | Power up the scanner head and start scanning only after power up |
| **Parameters** | Empty |
| **Return Value** | Empty |
| **Attention** | **1. PDA default power on the bright screen to open the scanning head, off the screen to close the scanning head; PDA has handled the logic related to the scanning head switch.**  **2. Non-essential do not call, if you need to call can not be called frequently, the power switch takes time** |

###### 3.2. Close the scanning head

|  |  |
| --- | --- |
| **void close( )** | |
| **Description** | Close the scan head, after closing the scan head stops working and no longer responds to the decoding request |
| **Parameters** | Empty |
| **Return Value** | Empty |
| **Attention** | **1. PDA default power on the bright screen to open the scanning head, off the screen to close the scanning head; PDA has handled the logic related to the scanning head switch.**  **2. Non-essential do not call, if you need to call can not be called frequently, the power switch takes time** |

###### 3.3. Start scanning

|  |  |
| --- | --- |
| **void scan\_start( )** | |
| **Description** | Start to scan the code, after the call the scanning head will come out of the light and identify the barcode |
| **Parameters** | Empty |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.4. Stop scanning

|  |  |
| --- | --- |
| **void scan\_stop( )** | |
| **Description** | Stop sweeping the code, call after the scan head off are extinguished |
| **Parameters** | Empty |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.5、Lock the scan button

|  |  |
| --- | --- |
| **void lockScanKey(boolean enable)** | |
| **Description** | Whether to enable the "scan button" (usually the yellow button) to trigger the scan operation |
| **Parameters** | **enable , whether to enable the scan button**  **true :** Use the "scan button" to trigger a scan operation **(default)**  **false :** Disable the "scan button" to trigger the scan operation |
| **Return Value** | Empty |
| **Attention** | 1. **If the scan button is disabled, the scan button cannot be pressed to scan** 2. **If you need to use the original scan key for other functions, such as UHF, temperature measurement, etc., you can disable the scan key and use onKeyDown(), onKeyUp() and other listening methods to listen to the key value and do other functions.** 3. **The software needs to be re-enabled before exiting the scan button to avoid not triggering the scan function properly** |

###### 3.6. Configure the output method of scanning results

|  |  |
| --- | --- |
| **void setOutputMode(int mode)** | |
| **Description** | Configure the scan result output mode |
| **Parameters** | **mode => 0:** focus output, if there is an edit box focus output to the edit box, if there is no focus will trigger the UI interface controls; **(default)**  **1:** Broadcast mode sending, you can listen to the scan result through Android broadcast mechanism.  The default broadcast is defined as follows.  **action：android.intent.action.SCANRESULT**  **extras: value The** result of a scan, the value of a failed scan is null  **length The length of the** barcode, the length of the failed scan is 0  **2:** Simulation of key output, simulating the output effect of human manually triggered keys  **3:** Paste the scan result to simulate the effect of copy and paste |
| **Return Value** | Empty |
| **Attention** | **You need to register to listen to action broadcasts before you can receive scan broadcasts** |

###### 3.7、Configuration of scanning beep, vibration, indicator

|  |  |
| --- | --- |
| **void enablePlayBeep(boolean enable)** | |
| **Description** | Whether to play sound after successful scanning |
| **Parameters** | **enable => true :** play sound after successful scan **(default)**  **false :** No sound is played after successful scanning |
| **Return Value** | Empty |

|  |  |
| --- | --- |
| **void enableFailurePlayBeep(boolean enable)** | |
| **Description** | Whether to play sound after scan failure |
| **Parameters** | **enable => true :** play sound after scan failure  **false :** No sound is played after a failed scan **(default)** |
| **Return Value** | Empty |

|  |  |
| --- | --- |
| **void enablePlayVibrate(boolean enable)** | |
| **Description** | Does it vibrate after a successful scan |
| **Parameters** | **enable => true :** Vibrate after successful scanning  **false :** No vibration after successful scanning **(default)** |
| **Return Value** | Empty |
| **Attention** | None |

|  |  |
| --- | --- |
| **void lightSet(boolean enable)** | |
| **Description** | Configure scan indicator status |
| **Parameters** | **enable => true :** Scan process on indicator  **false :** Scan process off indicator **(default)** |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.8. Additional terminator

|  |  |
| --- | --- |
| **void enableAddKeyValue(int value)** | |
| **Description** | Append the specified keystroke value to the scan result |
| **Parameters** | **value => 0:** no content attached  **1:** Additional carriage return **(default)**  **2:** Additional TAB key  **3:** Additional line break(\n) |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.9. Data processing rules

|  |  |
| --- | --- |
| **void addPrefix(String text)** | |
| **Description** | After a successful scan, add a prefix to the scan result string |
| **Parameters** | **text =>** prefix string added to the scan result |
| **Return Value** | Empty |
| **Attention** | None |

|  |  |
| --- | --- |
| **void addSuffix(String text)** | |
| **Description** | After a successful scan, add a suffix to the scan result string |
| **Parameters** | **text =>** suffix string added to the scan result |
| **Return Value** | Empty |
| **Attention** | None |

|  |  |
| --- | --- |
| **void filterCharacter(String text)** | |
| **Description** | Filter specific characters |
| **Parameters** | **text =>** filter specific characters |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.10, configure the trigger method

|  |  |
| --- | --- |
| **void continuousScan(boolean enable)** | |
| **Description** | Configuring Continuous Scan Status |
| **Parameters** | **enable =>** true: open continuous scanning mode (trigger scan button to start continuous scanning/stop continuous scanning)  false: close the continuous sweep mode |
| **Return Value** | Empty |
| **Attention** | **After opening the continuous scanning, you need to trigger the start of the scan before entering the continuous scanning state, and then trigger a scan to stop the continuous scanning state** |

|  |  |
| --- | --- |
| **void effortScan(boolean enable)** | |
| **Description** | Power saving mode, press the button until the barcode is solved or the scan timeout will not stop scanning |
| **Parameters** | **enable =>** true: turn on power saving mode  false: turn off power-saving mode |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.11, scan timeout time

|  |  |
| --- | --- |
| **void setTimeOut(int value)** | |
| **Description** | Set the scan timeout time and stop scanning automatically after the specified time |
| **Parameters** | **value =>** Scan interval time, valid range 1000~10000 ms |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.12, continuous scanning interval

|  |  |
| --- | --- |
| **void setIntervalTime(int value)** | |
| **Description** | Set the timeout time and stop scanning automatically after the specified time |
| **Parameters** | **value =>** Continuous scan interval time, valid range 100~3000 ms |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.13. Set the character editing format

|  |  |
| --- | --- |
| **void setEncodeFormat(int value)** | |
| **Description** | Configure decoding character encoding format |
| **Parameters** | **mode => 0: Auto (default)**  **1: GB2312**  **2: GBK**  **3：GB18030**  **4: UTF-8**  **5: ISO-8859-1**  **6: BIG5**  **7：SJIS**  **8：EUC-JP** |
| **Return Value** | Empty |
| **Attention** | **You can change the decoding character encoding format if there is garbled code** |

###### 3.14、Whether to delete the existing content of the edit box

|  |  |
| --- | --- |
| **void setDelete(boolean enable)** | |
| **Description** | Configure decoding character encoding format |
| **Parameters** | **enable => true :** Delete the content of the edit box after the barcode is scanned  **false:** do not delete the content of the edit box after scanning the barcode |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.15. Restore default configuration parameters

|  |  |
| --- | --- |
| **void resetScan( void )** | |
| **Description** | Configure the decoding character encoding format to restore the default settings of scanning parameters, and restore the scanning configuration parameters to the factory state after restoration |
| **Parameters** | None |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.16, decoding mode

|  |  |
| --- | --- |
| **void setCenterMode( int mode )** | |
| **Description** | Configure decoding region mode |
| **Parameters** | **mode** ,set decoding mode  **0**: Region decoding **(default)**  **1**: Central decoding |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.17、Laser fill light mode

|  |  |
| --- | --- |
| **void setAimLightMode( int mode )** | |
| **Description** | Set laser fill light mode |
| **Parameters** | **mode** ,laser light configuration mode.  **0**: fill light and laser are on **(default)**  **1**:Only open laser  **2**:Only open fill light  **3**: fill light and laser are off |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.18、Configuration of barcode control switch

|  |  |
| --- | --- |
| **void setBarcodeEnable( int barcodeId,boolean enable)** | |
| **Description** | Set laser fill light mode |
| **Parameters** | **barcodeId** ,barcode type, supports the following values of type.  0: Aztec  1: Codabar  2: Code11  3: Code128  4: Code39  6: Code93  8: DataMatrix  9: EAN8  10: EAN13  11: Interleaved 2 of 5  12: Maxicode  13: Micropdf  15：Pdf417  17: QR  19: UPCA  20: UPCE0  48: HANXIN  **enable**, barcode status, supports the following values.  true: Support specified barcode decoding  false: turn off the specified barcode decoding |
| **Return Value** | Empty |
| **Attention** | None |

###### 3.19. Picture output mode

|  |  |
| --- | --- |
| **void setImageMode( int mode )** | |
| **Description** | Setting the TUP mode |
| **Parameters** | **mode** ,light mode.  **0**:off **(default)**  **1**:Always output |
| **Return Value** | Empty |
| **Attention** | **This interface is only available for DS7000/7000Pro scan engines** |

###### 3.20、Multi-barcode enable

|  |  |
| --- | --- |
| **void setMultiBarcodeEnable( boolean mode )** | |
| **Description** | Set multi-barcode switch |
| **Parameters** | **enable**, multi-barcode switch.  **true**: open  **false**: close |
| **Return Value** | Empty |
| **Attention** | **This interface is only available for DS7000/7000Pro scan engines** |

###### 3.21、Configure multiple barcode precision mode

|  |  |
| --- | --- |
| **void setMultiBarcodePreciseStatus( boolean mode )** | |
| **Description** | Configure multi-barcode precision mode, that is, if the number of configured multi-barcodes is 3 and this configuration is on, then there must be three barcodes to decode successfully |
| **Parameters** | **enable**, multi-barcode precision mode switch.  **true**: open  **false**: close |
| **Return Value** | Empty |
| **Attention** | **This interface, which needs to be turned on for multi-barcode mode to make sense** |

###### 3.22、Configure the number of multiple barcodes

|  |  |
| --- | --- |
| **void setMultiBarcodeNumber( int number )** | |
| **Description** | Configure the number of multiple barcodes identifiable |
| **Parameters** | number, multi-barcode number: 1-20 |
| **Return Value** | Empty |
| **Attention** | **This interface, which needs to be turned on for multi-barcode mode to make sense** |

###### 3.23. Registering scan result listener objects

|  |  |
| --- | --- |
| **void registerScan( IScanListener mIScanListener )** | |
| **Description** | Register a listener and implement a listener callback method to get the path to the saved image |
| **Parameters** | mIScanListener ,listener object ([[3.25]](#_3.25、回调接口类) with detailed description) |
| **Return Value** | Empty |
| **Attention** | **This interface is registered once at the beginning of an app's lifecycle (onCreate) and logged out at the end of the lifecycle (onDestroy)[[3.24]](#_3.24、注销扫描结果监听对象)** |

###### 3.24. Logout of the scan result listener object

|  |  |
| --- | --- |
| **void unregisterScan( IScanListener mIScanListener )** | |
| **Description** | Register a listener and implement the listener callback method to get the path to the saved image |
| **Parameters** | mIScanListener ,listener object ([[3.25]](#_3.25、回调接口类) with detailed description) |
| **Return Value** | Empty |
| **Attention** | **This interface is registered with[[3.23]](#_3.23、注册扫描结果监听对象) After success, wait for the end of the APP life cycle (onDestroy) to logout** |

###### 3.25, callback interface class

|  |  |
| --- | --- |
| **IScanListener** | |
| **Description** | Scanning data callback interface, can be used to receive scanning data information |
| **Usage** | IScanListener miScanListener = new IScanListener() {  @Override  public void onScanResults(String data, int type, long decodeTime,  long keyDownTime,String imagePath) {    }  }; |
| **Parameters** | data Scan data (scan failed: (data == null || data.isEmpty()))  type barcode type (see the table of barcode types on the attached page)[[4.1]](#_4.1、条码类型对照表) )  decodeTime Scan time  keyDownTime The time when the key is pressed  imagePath image storage address (need to be turned on first to save images to be valid) |
| **Attention** | 1. **This interface data callback and "scan result output mode"[[3.6]](#_3.6、配置扫描结果的输出方式) exist at the same time, i.e. when it is in focus output mode, the scan result will be fed back to the decoding result in both focus output and callback interface (it is recommended to set the output mode to broadcast when using the callback interface)** 2. **onScanResults callback method can not update the UI, use runOnUiThread () to update the UI, see the demo example for details** |

#### 4、Attachment page

###### 4.1、Barcode type comparison table

**Description**: Code Type ID is int type

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbology**  **(Barcode type)** | **Code Type ID**  **(Barcode type ID corresponding to different scanning heads )** | | |
| **Zebra** | **idata** | **Honey Well** |
| **Code 39** | **1** | **'b'** | **'b'** |
| **Codebar** | **2** | **'a'** | **'a'** |
| **Code 128** | **3** | **'j'** | **'j'** |
| **Interleaved 2 of 5** | **6** | **'e'** | **'e'** |
| **Code 93** | **7** | **'i'** | **'i'** |
| **UPC-A** | **8** | **'c'** | **'c'** |
| **UPC-E** | **9** | **'E'** | **'E'** |
| **EAN-8** | **10** | **'D'** | **'D'** |
| **EAN-13** | **11** | **'d'** | **'d'** |
| **Code 11** | **12** | **'h'** | **'h'** |
| **GS1-128** | **15** | **'j'** | **'j'** |
| **PDF417** | **17** | **'r'** | **'r'** |
| **ISBT 128** | **25** | **'j'** | **'j'** |
| **MicroPDF417** | **26** | **'R'** | **'R'** |
| **Data Matrix** | **27** | **'w'** | **'w'** |
| **QR Code** | **28** | **'s'** | **'s'** |
| **Maxicode** | **37** | **'x'** | **'x'** |
| **Aztec** | **45** | **'z'** | **'z'** |
| **Han Xin** | **183** | **'H'** | **'H'** |
| **EAN-128** |  | **'I'** | **'I'** |
| **Code 49** |  | **'l'** | **'l'** |
| **IATA 2 of 5** |  | **'F'** | **'F'** |
| **standerd2of5** |  | **'f'** | **'f'** |
| **Matrix2of5** |  | **'m'** | **'m'** |