Newland AIDC
SCANNING MADE SIMPLE

NEWLAND ANDROID PDA

Mobile Computers

Version: V1.07

API Handbook

Disclaimer

Please read the manual carefully before using the product and operate it according to the manual. It is advised that you keep this manual for future reference.

Do not disassemble the device or remove the seal label from the device; doing so will void the product warranty provided by Newland Europe BV.

All pictures in this manual are for reference only, and the actual product may differ.

Regarding product modification and update, Newland Europe BV reserves the right to make changes to any software or hardware to improve reliability, function, or design at any time without notice. The information contained herein is subject to change without prior notice.

The products depicted in this manual may include software copyrighted by Newland Europe BV or a third party. The user, corporation or individual shall not duplicate, in whole or in part, distribute, modify, decompile, disassemble, decode, reverse engineer, rent, transfer, or sublicense such software without prior written consent from the copyright holders.

This manual is copyrighted. No part of this publication may be reproduced, distributed, or used in any form without Newland Europe BV's written permission.

Risk Warning Regarding Unauthorized System Updates:
You should use the Newland-provided tool to update this product's system. Modifying system files by installing a third-party ROM system or using any cracking method may result in product malfunction or data loss and void your warranty.

Newland Europe BV reserves the right to make a final interpretation of the statement above.

Newland Europe BV

Rolweg 25, 4104 AV, Culemborg,
The Netherlands
www.newland-id.com

Newland Europe BV is a subsidiary of Newland Digital Technology Co., Ltd. Our general conditions of Purchase, Sale and Delivery are filed with the Record Office of the Chamber of Commerce of Utrecht, The Netherlands.

K.v.K. H.R. Utrecht / Chamber of Commerce Utrecht: Reg. nr. 17109876

Revision History

Version	Description	Date
V1.0.0	Initial release.	January 16, 2018
V1.0.1	Updated the “Change the Scanner Settings” and “Reserved Keys” sections, and added the “Appendix” section.	June 19, 2018
V1.0.2	Added the “Configuring Symbologies” section.	March 25, 2019
V1.03	Updated the “Scan Barcode” and “Stop Scanning” sections	May 31, 2019
V1.04	Updated the “Configuring Scanner Parameters” section	Step.9, 2020
V1.05	Added raw data interface for scan result byte. Added settings for NFC, positioning, soft keyboard and APN.	Jun.6,2022
V1.06	Added the “Enable or Disable Recent Apps” section Added the “Delay Mode” option Added the “Send Scan Fail Broadcast” option Added the “DOTCODE” symbology Added the “Advanced Settings” section.	Jun.29,2022
V1.07	Updated the “EXTRA_SCAN_SETTINGS_RESTORE” section Updated the “EXTRA_SCAN_AUTOENT” section. Updated the “EXTRA_TRIG_MODE” section. Updated the default setting of “SCAN_ENCODE”. Deleted the table of “programmable barcode parameters”. Deleted the table of Advanced Settings.	April.6, 2023

Table of Contents

About This Manual

This manual is applicable to Newland Android Portable Data Collectors (hereinafter referred to **"the terminal"**).

Development Environment

All APIs are built based on standard Android broadcast mechanism, so there is no need for additional SDKs. The terminal application development environment is the same as Android application development environment.

Obtain Product Model Number

To get the product model number, use **android.os.Build.MODEL**. According to this, the application can adapt to manufacturers' different devices, such as MT65 and MT90.

Barcode Scanner

Scan Barcode

To activate the terminal to scan barcode, application should send the following broadcast to the system.

- Broadcast: **nlscan.action.SCANNER_TRIG**

  To trigger the scan engine.

- Extra scan timeout parameter: **SCAN_TIMEOUT** (value: int, 1-9; default value: 3; unit: second)

  To set scan timeout, i.e. the maximum time a scan attempt can last.

- Extra scan type parameter: **SCAN_TYPE** (value: 1 or 2; default value: 1)

  To set scan type: Value = 1, read one barcode during a scan attempt

  Value = 2, read two barcodes during a scan attempt (This feature is **NOT** available)

Example 1:

```
Intent intent = new Intent ("nlscan.action.SCANNER_TRIG");
mContext.sendBroadcast(intent);
```

Example 2:

```
Intent intent = new Intent ("nlscan.action.SCANNER_TRIG");
intent.putExtra("SCAN_TIMEOUT", 4);// SCAN_TIMEOUT value: int, 1-9; unit: second
intent.putExtra("SCAN_TYPE ", 2);// SCAN_TYPE: read two barcodes during a scan attempt
mContext.sendBroadcast(intent);
```

Note: When a scan and decode session is in progress, sending the broadcast above will stop the ongoing session. When scanning barcode by pressing the Scan key, it is processed at the bottom layer, thus application does not need to listen for Scan KeyPress event or send the broadcast.

Get Barcode Data

There are three ways to get barcode data:

1. Fill in EditText directly: Output scanned data at the current cursor position in EditText.

2. Simulate keystroke: Output scanned data to keyboard buffer to simulate keyboard input and get the data at the current cursor position in TextBox.

3. Output via API: Application acquires scanned data by registering a broadcast receiver and listening for specific broadcast intents.

- Broadcast: **nlscan.action.SCANNER_RESULT**
  To get barcode data.
- Extra scan result 1 parameter: **SCAN_BARCODE1**
  To get the data of barcode 1.
  Type: String
- Extra scan result 1 raw byte parameter: **scan_result_one_bytes**
  To get the byte data of barcode 1.
  Type: byte[]
- Extra scan result 2 parameter: **SCAN_BARCODE2**
  To get the data of barcode 2.
  Type: String
- Extra scan result 2 raw byte parameter: **scan_result_two_bytes**
  To get the byte data of barcode 2.
  Type: byte[]
- Extra symbology ID number parameter: **SCAN_BARCODE_TYPE**
  Type: int (-1 indicates failure to get symbology ID Number)
  To get the ID number of the barcode scanned (Refer to the “Symbology ID Number” table in

Appendix to get the barcode type).

- Extra scan state parameter: **SCAN_STATE** (value: fail or ok)
  To get the status of scan operation: Value = fail, operation failed
  Value = ok, operation succeeded
  Type: String

Example:

Register broadcast receiver:

```
mFilter= newIntentFilter("nlscan.action.SCANNER_RESULT");
mContext.registerReceiver(mReceiver, mFilter);
```

Unregister broadcast receiver:

```
mContext.unregisterReceiver(mReceiver);
```

Get barcode data:

```
mReceiver= newBroadcastReceiver() {
        @Override
publicvoidonReceive(Context context, Intent intent) {
            final String scanResult_1=intent.getStringExtra("SCAN_BARCODE1");
            final String scanResult_2=intent.getStringExtra("SCAN_BARCODE2");
                // Raw byte data of the scan result
            final byte[] scanResultByte_1=intent. intent.getByteArrayExtra("scan_result_one_bytes");
            final byte[] scanResultByte_2= intent. intent.getByteArrayExtra("scan_result_two_bytes");
            final int barcodeType = intent.getIntExtra("SCAN_BARCODE_TYPE", -1); // -1:unknown
            final String scanStatus=intent.getStringExtra("SCAN_STATE");
                if("ok".equals(scanStatus)){
                //Success
                }else{
                //Failure, e.g. operation timed out
                }
    }
    };
```

Stop Scanning

Note: When scanning barcode by pressing the Scan key, it is processed at the bottom layer to stop the scan session, thus application does not need to send the broadcast. Even if you scan barcode by pressing the Scan key, application only need to acquire scanned data by registering a broadcast receiver and listening for specific broadcast intents, without having to send the broadcast to activate and stop scanning.

Use the broadcast **nlscan.action.STOP_SCAN** to stop an ongoing decode session.

Example:

```
Intent stopIntent = new Intent("nlscan.action.STOP_SCAN");
mContext.sendBroadcast(stopIntent);
```

Change the Scanner Settings

Configuring Scanner Parameters

Application can set one or more scanner parameters, such as enable/disable scanner, by sending to the system the broadcast **ACTION_BAR_SCANCFG** which can contain up to 3 parameters.

Parameter	Type	Description (* indicates default)
EXTRA_SCAN_POWER	INT	Value = 0 Disable scanner = 1 Enable scanner* Note: When scanner is enabled, it will take some time to initialize during which all scan requests will be ignored.
EXTRA_TRIG_MODE	INT	Value = 0 Level mode = 1 Continuous mode = 2 Pulse mode* = 4 Delay mode (Press and hold the scan trigger to aim at barcode then release it to start a decode session which continues until the decode session timeout expires or a barcode is decoded. It is advised to use this scan mode and the Acuscan Decoding feature to ensure that only the desired barcodes are read if multiple barcodes are placed closely together.)
EXTRA_SCAN_MODE	INT	Value = 1 Fill in EditText directly* = 2 Simulate keystroke = 3 Output via API
SEND_SCAN_FAIL_BROADCAST	INT	Value = 0 Disable the send scan fail braoadcast = 1 Enable the send scan fail braoadcast*
EXTRA_SCAN_AUTOENT	INT	Value = 0 Do not add a line feed* = 1 Add a line feed Send an Enter Key after each barcode is scanned.
EXTRA_SCAN_NOTY_SND	INT	Value = 0 Sound notification off = 1 Sound notification on*
EXTRA_SCAN_NOTY_VIB	INT	Value = 0 Vibration notification off* = 1 Vibration notification on
EXTRA_SCAN_NOTY_LED	INT	Value = 0 LED notification off = 1 LED notification on*
SCAN_TIMEOUT	LONG	Set decode session timeout (millisecond) Value = 0-9000; default: 3000*
SCAN_INTERVAL	LONG	Set timeout between decode sessions (millisecond)

		Value >= 50; default: 500*
TRIGGER_MODE_MAIN	INT	Value = 0 Disable the Scan key on front panel as scan trigger = 1 Enable the Scan key on front panel as scan trigger*
TRIGGER_MODE_LEFT	INT	Value = 0 Disable the Scan key on left side as scan trigger = 1 Enable the Scan key on left side as scan trigger*
TRIGGER_MODE_RIGHT	INT	Value = 0 Disable the Scan key on right side as scan trigger = 1 Enable the Scan key on right side as scan trigger*
TRIGGER_MODE_BLACK	INT	Value = 0 Disable the trigger on pistol grip as scan trigger = 1 Enable the trigger on pistol grip as scan trigger* (Precondition: The terminal supports this feature)
NON_REPEAT_TIMEOUT	LONG	Set reread delay (millisecond) Value = 0 Reread same barcode with no delay* > 0 Do not allow to reread same barcode before the delay expires
SCAN_PREFIX_ENABLE	INT	Value = 0 Disable prefix = 1 Enable prefix*
SCAN_SUFFIX_ENABLE	INT	Value = 0 Disable suffix = 1 Enable suffix*
SCAN_PREFIX	STRING	Set prefix Value = Hexadecimal value of prefix character; default: null* e.g. 0x61 should be entered as 61.
SCAN_SUFFIX	STRING	Set suffix Value = Hexadecimal value of suffix character; default: null* e.g. 0x61 should be entered as 61.
SCAN_ENCODE	INT	Character encoding Value = 1 UTF-8 = 2 GBK = 3 ISO-8859-1 = 4 AUTO* = 5 Other Should enter the value of SCAN_OTHER_ENCODE at the same time = 6 windows-1251
OUTPUT_RECOVERABLE	BOOLEAN	Value = true Enable overwrite output = false Disable overwrite output*
EXTRA_OUTPUT_EDITOR_ACTION_ENABLE	INT	Value = 0 Disable software key event output * = 1 Enable software key event output
EXTRA_OUTPUT_EDITOR_ACTION	INT	Value = 0 IME_ACTION_UNSPECIFIED = 1 IME_ACTION_NONE = 2 IME_ACTION_GO = 3 IME_ACTION_SEARCH = 4 IME_ACTION_SEND = 5 IME_ACTION_NEXT = 6 IME_ACTION_DONE * = 7 IME_ACTION_PREVIOUS
BROADCAST_OUTPUT_ACTION	STRING	Broadcast output settings Action value

BROADCAST_OUTPUT_EXTRA_KEY_RESULT_1	STRING	Broadcast output settings Barcode Result 1 parameter
BROADCAST_OUTPUT_EXTRA_KEY_RESULT_2	STRING	Broadcast output settings Barcode Result 2 parameter
BROADCAST_OUTPUT_EXTRA__KEY_BARCODE_TYPE	STRING	Broadcast output settings Barcode type parameter
BROADCAST_OUTPUT_EXTRA__KEY_BARCODE_TYPE_NAME	STRING	Broadcast output settings Barcode type name parameter
EXTRA_SCAN_SETTINGS_RESTORE	BOOLEAN	Value = true Restore the default settings

Example 1: Disable scanner

```
Intent intent = new Intent ("ACTION_BAR_SCANCFG");
intent.putExtra("EXTRA_SCAN_POWER", 0);
mContext.sendBroadcast(intent);
```

Example 2: Output via API, add a line feed

```
Intent intent = new Intent ("ACTION_BAR_SCANCFG");
intent.putExtra("EXTRA_SCAN_MODE", 3);
intent.putExtra("EXTRA_SCAN_AUTOENT", 1);
mContext.sendBroadcast(intent);
```

Configuring Symbologies

Application can set barcode parameter, such as enable/disable a symbology, transmit check character, set minimum/maximum length by sending to the system the broadcast **ACTION_BARCODE_CFG** which contains the following three parameters.

Parameter	Type	Description
CODE_ID	STRING	Value = Barcode type e.g. ”CODE128”
PROPERTY	STRING	Value = Barcode parameter e.g. ”Enable”, “Minlen”, or “TrsmtChkChar”
VALUE	STRING	Value = Value of the barcode parameter e.g. To enable a symbology, set the value to “1”

Example: Transmit EAN-8 check character

```
Intent intent = new Intent ("ACTION_BARCODE_CFG");
intent.putExtra("CODE_ID", “EAN8”);
intent.putExtra("PROPERTY", “TrsmtChkChar”);
intent.putExtra("VALUE", “1”); // “1” Enable EAN-8, ”0” Disable EAN-8
mContext.sendBroadcast(intent);
```

Reserved Keys

The terminal provides reserved keys, for example:

MT90 provides one reserved key: F6.

MT65 provides four reserved keys: F1、F2、F3、F4.

Application can define reserved key’s functions as per actual needs

Example 1: Process the KeyDown event of reserved key

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
switch (keyCode)
        {
case KeyEvent.KEYCODE_F6:
showInfo("F6 KeyDown\n");
break;
        }
return super. onKeyDown(keyCode,event);
}
```

Example 2: Process the KeyUp event of reserved key

```
public boolean onKeyUp(int keyCode, KeyEvent event) {
switch (keyCode)
        {
case KeyEvent.KEYCODE_F6:
showInfo("F6 KeyUp\n");
break;
        }
return super.onKeyDown(keyCode, event);
    }
```

Other APIs

Notification Bar Pull-down

To enable/disable the notification bar pull-down, application should send to the system the broadcast **nlscan.action.STATUSBAR_SWITCH_STATE** with the value of Extra parameter ENABLE set to be true/false.

Example: Disable the notification bar pull-down

```
Intent intent = new Intent("nlscan.action.STATUSBAR_SWITCH_STATE");
intent.putExtra("ENABLE", false);
context.sendBroadcast(intent);
```

Press the Home Key to Switch to Desktop

To enable/disable the feature of switching to desktop by pressing the Home key, application should send to the system the broadcast **nlscan.action.HOMEKEY_SWITCH_STATE** with the value of Extra parameter ENABLE set to be true/false.

Example: Disable the feature of switching to desktop by pressing the Home key

```
Intent intent = new Intent("nlscan.action.HOMEKEY_SWITCH_STATE");
intent.putExtra("ENABLE", false);
context.sendBroadcast(intent);
```

Set the System Time

To set the system time, application should send to the system the broadcast **nlscan.action.SET_TIME** with the value of Extra parameter TIME_MS set to be a string represented as the number of millisecond.

Example:

```
Public long getTimeMillis(){
    Calendar c=Calendar.getInstance();
    c.set(2016,0,1,0,0,0);
    return c.getTimeInMillis();
}
Intent it = new Intent("nlscan.action.SET_TIME");
long mills = getTimeMillis();
it.putExtra("TIME_MS", String.valueOf(mills));
mContext.sendBroadcast(it);
```

Set the NFC, Positioning, Soft Keyboard, and APN

Application can set NFC, Positioning, Soft Keyboard, and APN by sending to the system the broadcast **com.nlscan.action.backuprecovery** which contains the following parameters.

Parameter	Type	Description
Set	STRING	Json String

Calling Example:

```
String json = "{\n" +
                "\t\"quick_setting\": [{\n" +
                "\t\t\"quick_setting\": [{\n" +
                "\t\t\t\"NFC.Enable\": \"1\"\n" +
                "\t\t}],\n" +
                "\t\t\"set_data_diff_flag\": \"0\"\n" +
                "\t}],\n" +
                "\t\"version\": \"V0.00.001\"\n" +
                "}";
String action = "com.nlscan.action.backuprecovery";
String pkg = "com.nlscan.nlsbackuprecovery";
Intent intent = new Intent(action);
intent.setPackage(pkg);
intent.putExtra("set", json);
sendBroadcast(intent);
```

Json Explanation

NFC:

```
{
    "quick_setting": [
        {
            "quick_setting": [
                {
                    "NFC.Enable": "1"     //1：Enable   0：Disable
                }
            ],
```

```
            "set_data_diff_flag": "1"
        }
    ],
    "version": "V0.00.001"
}
```

Soft Keyboard:

```
{
    "quick_setting": [
        {
            "quick_setting": [
                {
                    "SHOWSOFTINPUT.Enable": "1"     //1：Enable   0：Disable
                }
            ],
            "set_data_diff_flag": "1"
        }
    ],
    "version": "V0.00.001"
}
```

Positioning:

```
{
    "device_setting": [
        {
            "start_intent": [
                {
                    "Intent.list": [
                        {
                            "type": "broadcast",
                            "action": "nlscan.action.WRITE_SETTINGS_DB",
                            "group_split_char":";",
                            "params":
"es-db-secure;es-name-location_providers_allowed;es-value-+network,gps;es-type-string"
                        }
```

```
                        ]
                    }
                ],
                "set_data_diff_flag": "1"
            }
        ]
        "version": "V0.00.001"
}
```

APN:

```
{
    "device_setting": [
        {
                        "apn": [
                {
                    "RESET_APN.Enable": "1",
                    "APN_LIST.list": [
                        {
                            "APN_PROXY": "",                    //proxy
                            "APN_TYPE": "",                     //type
                            "APN_SUBID": "1",                   //SIM card id, "0" or "1" for single SIM
 card
                            "APN_MVNO_TYPE": "",    //MVNO Type
                            "APN_MMSC": "",                                 //MMSC
                            "APN_MVNO_VALUE": "",  //MVNO Value
                            "APN_AUTHTYPE": "",                 // Auth type, optional value:
PAP,CHAP,PAP OR CHAP
                            "APN_SERVER": "",                   //Server
                            "APN_APN": "11111",                 //APN
                            "APN_USER": "",                     //User Name
                            "APN_PROTOCOL": "IPv4/IPv6",      //Protocol, optional value：IPv4，IPv6，
IPv4/IPv6
                            "APN_NAME": "11111",      //APN name
                            "APN_PASSWORD": "",                 //Password
                            "APN_PORT": "",                     //Port
                            "APN_OPERTYPE": "2",    // Do not change this item
                            "APN_MMSPROXY": "",                 //MMS proxy
```

```
                                "APN_ROAMING_PROTOCOL": "IPv4/IPv6",   // Roaming protocol，
optional value：IPv4，IPv6，IPv4/IPv6
                                "APN_MMSPORT": "",                  //MMS port
                                "APN_BEARER": ""                    //Bearer system
                            }
                        ]
                    }
                ],
                "set_data_diff_flag": "1"
            }
        ],
        "version": "V0.00.001"
}
```

Remarks for APN settings: Manually add APN on the terminal and verify that the function is normal. Then compelete the json parameter according to the detailed parameter interface of the APN newly added.

The above json can be set individually or in combination at one time as follows:

```
{
    "device_setting": [
        {
            "start_intent": [
                {
                    "Intent.list": [
                        {
                            "type": "broadcast",
                            "action": "nlscan.action.WRITE_SETTINGS_DB",
                            "group_split_char":";",
                            "params":
"es-db-secure;es-name-location_providers_allowed;es-value-+network,gps;es-type-string"
                        }
                    ]
                }
            ],
            "apn": [
                {
```

```
                    "RESET_APN.Enable": "1",
                    "APN_LIST.list": [
                        {
                            "APN_PROXY": "",
                            "APN_TYPE": "",
                            "APN_SUBID": "1",
                            "APN_MVNO_TYPE": "",
                            "APN_MMSC": "",
                            "APN_MVNO_VALUE": "",
                            "APN_AUTHTYPE": "",
                            "APN_SERVER": "",
                            "APN_APN": "11111",
                            "APN_USER": "",
                            "APN_PROTOCOL": "IPv4/IPv6",
                            "APN_NAME": "11111",
                            "APN_PASSWORD": "",
                            "APN_PORT": "",
                            "APN_OPERTYPE": "2",
                            "APN_MMSPROXY": "",
                            "APN_ROAMING_PROTOCOL": "IPv4/IPv6",
                            "APN_MMSPORT": "",
                            "APN_BEARER": ""
                        }
                    ]
                }
            ],
            "set_data_diff_flag": "1"
        }
    ],
    "quick_setting": [
        {
            "quick_setting": [
                {
                    "NFC.Enable": "1",
                    "SHOWSOFTINPUT.Enable": "1"
                }
            ],
            "set_data_diff_flag": "1"
        }
    ],
```

```
    "version": "V0.00.001"
}
```

Enable or Disable Recent Apps

Application can enable or disable the recent apps by sending to the system the broadcast **nlscan.action.SWITCH_RECENTS**

Example:

```
Intent intent = new Intent("nlscan.action.SWITCH_RECENTS");
intent.putExtra("ENABLE", false); //Disable the recent apps
context.sendBroadcast(intent);
```

Appendix

Symbology ID Number

ID Number	Symbology
0	ZASETUP
1	SETUP128
2	CODE128
3	UCCEAN128
4	AIM128
5	GS1_128
6	ISBT128
7	EAN8
8	EAN13
9	UPCE
10	UPCA
11	ISBN
12	ISSN
13	CODE39
14	CODE93
15	93I
16	CODABAR
17	ITF
18	ITF6
19	ITF14
20	DPLEITCODE
21	DPIDENTCODE
22	CHNPOST25
23	STANDARD25
23	IATA25
24	MATRIX25
25	INDUSTRIAL25
26	COOP25
27	CODE11
28	MSIPLESSEY
29	PLESSEY
30	RSS14

31	RSSLIMITED
32	RSSEXPANDED
33	TELEPEN
34	CHANNELCODE
35	CODE32
36	CODEZ
37	CODABLOCKF
38	CODABLOCKA
39	CODE49
40	CODE16K
41	HIBC128
42	HIBC39
43	RSSFAMILY
44	TriopticCODE39
45	UPC_E1
256	PDF417
257	MICROPDF
258	QRCODE
259	MICROQR
260	AZTEC
261	DATAMATRIX
262	MAXICODE
263	CSCODE
264	GRIDMATRIX
265	EARMARK
266	VERICODE
267	CCA
268	CCB
269	CCC
270	COMPOSITE
271	HIBCAZT
272	HIBCDM
273	HIBCMICROPDF
274	HIBCQR
275	DOTCODE
512	POSTNET
513	ONECODE
514	RM4SCC
515	PLANET

516	KIX
517	APCUSTOM
518	APREDIRECT
519	APREPLYPAID
520	APROUTING
768	NUMOCRB
769	PASSPORT
770	TD1
2048	PRIVATE
2049	ZZCODE
65535	UNKNOWN

SCANNING MADE SIMPLE

Newland AIDC EMEA
+31 (0) 345 87 00 33
info@newland-id.com

Rolweg 25
4104 AV Culemborg
The Netherlands