



# Newland

SCANNING MADE SIMPLE



**MT90 Orca**  
mobile computer  
SDK Handbook

## Revision History

Version	Description	Date
V1.0.0	Initial release.	January 16, 2018

## Table of Contents

<b>About This Manual</b> .....	<b>1</b>
<b>Development Environment</b> .....	<b>1</b>
<b>Obtain Product Model Number</b> .....	<b>1</b>
<b>Barcode Scanner</b> .....	<b>1</b>
Scan Barcode.....	1
Get Barcode Data .....	2
Stop Scanning.....	3
Change the Scanner Settings.....	3
<b>Reserved Keys</b> .....	<b>4</b>
<b>Other APIs</b> .....	<b>5</b>
Expand the Status Bar .....	5
Press the Home Key to Switch to Desktop.....	5
Set the System Time.....	5

## About This Manual

This manual is applicable to NLS-MT90 portable data collectors (hereinafter referred to as “**the MT90**” or “**the terminal**”).

## Development Environment

All APIs are built based on standard Android broadcast mechanism, so there is no need for additional SDKs. The MT90 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`, for example, MT90.

## Barcode Scanner

### Scan Barcode

To activate the MT90 to scan barcode, application should send the following broadcast to the system.

- Broadcast: `nlsan.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 ("nlsan.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 2 parameter: SCAN\_BARCODE2  
To get the data of barcode 2.  
Type: String
  - 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");  
        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

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

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*
EXTRA_SCAN_MODE	INT	Value = 1 Fill in EditText directly* = 2 Simulate keystroke = 3 Output via API
EXTRA_SCAN_AUTOENT	INT	Value = 0 Do not add a line feed* = 1 Add a line feed
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*

**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);
```

## Reserved Keys

The MT90 provides one reserved key F6. Application can define its function 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

### Expand the Status Bar

To set the status bar to be expandable/not expandable, 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:** Set the status bar to be not expandable

```
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);
```



# Newland

SCANNING MADE SIMPLE

#### **Newland EMEA HQ**

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

#### **D-A-C-H**

+49 (0) 6182 82916-16  
info@newland-id.de

#### **Benelux**

+31 (0) 345 87 00 33  
benelux@newland-id.com

#### **Italy**

+39 (0) 342 0562227  
italy@newland-id.com

#### **United Kingdom**

+44 (0) 1442 212020  
sales@newland-id.co.uk

#### **South Africa**

Gauteng: +27 (0) 11 553 8010  
Cape Town: +27 (0) 21 9140819  
info@newland-id.co.za

#### **Turkey**

+90 (0) 544 538 40 49  
turkey@newland-id.com

#### **France**

+39 (0) 345 8804096  
france@newland-id.com

#### **Ibérica**

+34 (0) 93 303 74 66  
info@newland-id.es

#### **Nordic & Baltic**

+46 (0) 70 88 47 767  
nordic@newland-id.com

#### **Russia**

+31 (0) 345 87 00 33  
russia@newland-id.com

#### **Middle East**

+39 (0) 345 8804096  
middleeast@newland-id.com

#### **Iran**

+90 (0) 544 538 40 49  
iran@newland-id.com