

# 标签打印机

---

NFC 功能说明

V1.4

**DothanTech 道臻技术**

2017 年 09 月 27 日

## 目 录

目 录.....	2
文档修改记录.....	3
1. NFC 功能概述.....	4
2. NFC 使用简述.....	4
3. NFC 功能说明.....	5
3.1 NFC Intent.....	5
3.2 onPrinterDiscovery 扫描到 NFC 打印机回调方法.....	5
3.3 NFC 功能使用说明.....	6
3.3.1 扫描到 NFC 打印机时的两种情况.....	6
3.3.2 使用 NFC 功能需要进行的相关设置.....	7

## 文档修改记录

序号	版本号	修改内容	修改者	修改日期
1	V0.1	文档建立	刘贇	2015-05-08
2	V0.9	文档完善	杨娟娟	2015-05-19
3	V1.0	文档发布	刘伟宏	2015-05-21
4	V1.1	增加 NFC 功能描述	刘贇	2015-05-22
5	V1.2	统一文档格式，正式发布	杨娟娟	2015-05-28
6	V1.3	修改 <a href="#">onPrinterDiscovery()</a> 的样例代码	胡殿兴	2017-04-05
7	V1.4	根据 NFCDemo 同步的更新样例代码	胡殿兴	2017-09-27

## 1. NFC 功能概述

NFC 功能说明文档，是针对扫描 NFC 打印机启动 App、获取 NFC 打印机信息提出的功能说明，方便用户在二次开发中调用功能接口，缩短开发周期，加快开发进度。

NFC 功能主要是指通过 NFC 扫描操作启动 Android App 并协助完成蓝牙标签打印机信息获取、设备配对、连接等一系列对接操作。NFC 功能集成于 IDzPrinter 接口中，接口提供了方便的回调通知机制，可在 NFC 功能扫描到打印机时自动使对应的 Android App 界面来到 Android 系统前台，并能够及时通知调用者，同时反馈该打印机信息；配合 IDzPrinter 接口的 connect 方法，调用者可根据实际情况进行打印机的配对、连接操作，或者当 App 正在另一打印机上进行打印时也可选择不进行连接，使标签打印机的对接操作更加灵活、便捷。

## 2. NFC 使用简述

NFC 功能的基本使用流程如下：

- 1) 在项目中引入 LPAPI-日期.jar 与 DzNFC-日期.jar；
- 2) 调用者首先需要在 Android App 的“AndroidManifest.xml”文件中进行相关设置，主要是添加蓝牙相关操作权限, NFC 相关操作权限和添加用于接收 NFC 系统广播消息的透明 Activity，详见 [3.3.2 章节](#)。
- 3) 调用者需要在 App 主界面 Activity 的 onCreate 方法中接收并解析扫描到 NFC 打印机时获取的 Intent 对象，通过是否解析到 NFC 扫描的打印机信息可判断 App 是 NFC 功能启动还是其它方式启动，由此决定是连接 NFC 扫描到的打印机还是连接上次的打印机，详见 [3.3.1 章节情况二](#)。

- 4) 调用者需要实现 IDzPrinter.IDzPrinterCallback 接口中的回调方法 [onPrinterDiscovery](#) 方法；在扫描到 NFC 打印机时，IDzPrinter 接口会调用该回调方法通知调用者，在回调方法中调用者可根据实际情况决定是否连接该打印机（例如当前正用另一打印机进行打印，可以选择不连接该打印机），详见 [3.3.1 章节情况一](#)。

## 3. NFC 功能说明

### 3.1 NFC Intent

在 Android App 还未启动时扫描 NFC，IDzPrinter 接口会启动 App，并将扫描到 NFC 打印机时获取的 Intent 对象作为 Parcelable 类型扩展对象附加在 App 主界面的启动 Intent 中（名称为"com.dothantech.manager.EXTRA\_NDEF\_INTENT"）传给 App 主界面。

### 3.2 onPrinterDiscovery 扫描到 NFC 打印机回调方法

使用 NFC 扫描功能，需实现 IDzPrinter.IDzPrinterCallback 接口中的回调方法 [onPrinterDiscovery](#)，在扫描到 NFC 打印机时，接口会调用该回调方法通知调用者。

回调方法描述如下：

功能简介	扫描到 NFC 打印机时，调用该回调方法。	
方法原型	<b>public void onPrinterDiscovery(PrinterAddress address, PrinterInfo info);</b>	
参数名称	address	扫描到的打印机地址对象。

	info	扫描到的打印机信息对象 ( IDzPrinter.PrinterInfo ) 。
返回值	无	
使用说明	此方法由 IDzPrinter 接口调用者实现，IDzPrinter 接口会在相应时刻调用该方法通知调用者。	
使用注意	此方法会在子线程中被调用，因此在此方法中无法进行一些 UI 界面操作或一些必须在 UI 线程中才可进行的操作。	

### 3.3 NFC 功能使用说明

#### 3.3.1 扫描到 NFC 打印机时的两种情况

情况一：当 Android App 已经启动，NFC 接口会使 App 界面回到 Android 系统主界面，如果在系统主界面初始化打印机的之后制定了相关回调函数，则系统主界面会调用回调函数中的 [onPrinterDiscovery](#)，调用者可在该方法中根据实际情况进行操作。回调方法中可如下操作：

##### 代码

```
private LPAPI mApi;           // 标签打印接口类，在OnCreate函数中创建
/**
 * 创建LPAPI实例的时候指定该回调接口，当NFC检测到打印机的时候会通过该回调接口的
 * onPrinterDiscovery()函数来通知客户,用户可以在改函数中连接打印机即可;
 */
private LPAPI.Callback callback = new LPAPI.Callback(){
    // 其它回调方法
    ...
    @Override
    public void onPrinterDiscovery (PrinterAddress address, PrinterInfo info) {
        // 打印过程中不能切换打印机;
        if (mApi.getPrinterState() != PrinterState.Printing) {
            mApi.openPrinterByAddress(arg0);    // 连接打印机
        }
    }

    // 其它回调方法
    ...
}
```

```
};
```

情况二：当 Android App 还未启动，NFC 接口会启动 App，并将扫描到 NFC 打印机时获取的 Intent 对象作为 Parcelable 类型扩展对象附加在 App 主界面的启动 Intent 中传给调用者（此 Parcelable 对象在 App 主界面启动 Intent 中的名称为 NFCManagerActivity.EXTRA\_NDEF\_INTENT）；调用者可调用 PrinterInfo.valueOf(Intent ndefIntent)方法解析此 Parcelable 数据，获得打印机信息对象。调用者在 App 主界面中可如下操作：

#### 代码

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // ----- 初始化打印接口 -----//
    this.mApi = LPAPI.Factory.createInstance(callback);
    // 如果应用主界面是通过打印机中的NFC来启动的，则可以通过字符串常量：
    NFCManagerActivity.EXTRA_NDEF_INTENT 来获取打印机信息，否则获取不到数据；
    Intent ndefIntent =
    getIntent().getParcelableExtra(NFCManagerActivity.EXTRA_NDEF_INTENT);
    PrinterInfo info = PrinterInfo.valueOf(ndefIntent);
    if (info != null) {
        this.mApi.openPrinterByAddress(info.getPrinterAddress()); // 连接打印机
    }
}
```

### 3.3.2 使用 NFC 功能需要进行的相关设置

- 使用 NFC 功能需要在 Android App 的 “AndroidManifest.xml” 文件中添加 NFC 相关操作权限，如下：

#### 代码

```
<!-- 蓝牙相关 -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

```
<uses-feature
    android:name="android.hardware.bluetooth_le"
    android:required="false" />

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<!-- NFC 相关操作权限 -->
<uses-permission android:name="android.permission.NFC" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.REORDER_TASKS" />
```

- 使用 NFC 功能需要在 Android App 的 “AndroidManifest.xml” 文件中添加 NFCManagerActivity 的 Activity 注册信息 ,同时需要配置该 Activity 的关联 NFC URL 过滤器 ( 可以为多个 URL ) , 如下 :

#### 代码

```
<!-- NFCManagerActivity 注册信息-->
<activity
    android:name="com.dothantech.nfcmanager.NFCManagerActivity"
    android:launchMode="singleInstance"
    android:theme="@android:style/Theme.NoDisplay" >
    <intent-filter>
        <action android:name="android.nfc.action.NDEF_DISCOVERED" />
        <category android:name="android.intent.category.DEFAULT" />
        <data
            android:host="www.dothantech.com"
            android:pathPrefix="/android/WeiDa.apk"
            android:scheme="http" />
        <data
            android:host="www.detonger.com"
            android:pathPrefix="/wdbq"
            android:scheme="http" />
    </intent-filter>
</activity>
```