添加NFC工厂测试模式

1. 解决思路：

由于NFC是通过在AndroidManifest.xml中注册接收信息的，因此如果NFC测试一直可用的话，就会导致每次有磁卡靠近时都会弹出NFC工厂测试应用。这是不允许的，因此在AndroidManifest.xml注册NFC工厂测试Activity时默认设置该Activity不可用，在进入NFC工厂测试前，设置NFC工厂测试模式为可用，在退出NFC工厂测试模式后设置NFC工厂测试模式为不可用。

2. 修改文件：



3. 在xml文件夹中创建元数据文件nfc\_tech\_filter.xml：

<?xml version="1.0" encoding="utf-8"?>

<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">

<tech-list>

<tech>android.nfc.tech.IsoDep</tech>

</tech-list>

<tech-list>

<tech>android.nfc.tech.NfcV</tech>

</tech-list>

<tech-list>

<tech>android.nfc.tech.NfcF</tech>

</tech-list>

</resources>

4. 创建NFCTest.java文件：

package com.bsm\_wqy.validationtools.sensor;

import android.app.Activity;

import android.app.PendingIntent;

import android.content.Intent;

import android.content.IntentFilter;

import android.content.IntentFilter.MalformedMimeTypeException;

import android.nfc.NfcAdapter;

import android.nfc.tech.IsoDep;

import android.nfc.tech.NfcF;

import android.nfc.tech.NfcV;

import android.os.Bundle;

import android.os.SystemProperties;

import android.util.TypedValue;

import android.util.Log;

import android.widget.TextView;

import android.view.WindowManager;

import android.content.ComponentName;

import android.content.pm.PackageManager;

import com.bsm\_wqy.validationtools.BaseActivity;

import com.bsm\_wqy.validationtools.R;

public class NFCTest extends BaseActivity {

private static final String TAG = "NFCTest";

private NfcAdapter mNfcAdapter;

private TextView mInfo;

private PendingIntent mPendingIntent;

private boolean mIsAutoTest = false;

private boolean mLastNfcOpen = false;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON,

WindowManager.LayoutParams.FLAG\_KEEP\_SCREEN\_ON);

SystemProperties.set("persist.sys.nfctest", "true");

setContentView(R.layout.activity\_nfc\_test);

setTitle(R.string.nfc\_test);

mInfo = (TextView) findViewById(R.id.info);

mNfcAdapter = NfcAdapter.getDefaultAdapter(this);

mPendingIntent = PendingIntent.getActivity(this, 0, new Intent(this,

getClass()).addFlags(Intent.FLAG\_ACTIVITY\_SINGLE\_TOP), 0);

if (mNfcAdapter == null) {

Log.e(TAG, "onCreate=>device not support nfc.");

finish();

return;

}

mLastNfcOpen = mNfcAdapter.isEnabled();

if (!mNfcAdapter.isEnabled()) {

Log.e(TAG, "onCreate=>nfc is disabled, please open nfc.");

mNfcAdapter.enable();

}

Intent intent = getIntent();

isFullTest = getIntent().getIntExtra("isFullTest", 0);

mIsAutoTest = (isFullTest != 0);

int fullTestActivityId = getIntent().getIntExtra("fullTestActivityId", 0);

setIsFullTest(isFullTest, ++fullTestActivityId);

onNewIntent(intent);

}

@Override

protected void onPause() {

super.onPause();

if (mNfcAdapter != null)

mNfcAdapter.disableForegroundDispatch(this);

}

@Override

protected void onResume() {

super.onResume();

if (mNfcAdapter != null) {

try {

mNfcAdapter.enableForegroundDispatch(this, mPendingIntent,

new IntentFilter[] { new IntentFilter(NfcAdapter.ACTION\_TECH\_DISCOVERED, "\*/\*") },

new String[][] { { IsoDep.class.getName() }, { NfcV.class.getName() },

{ NfcF.class.getName() }, });

} catch (MalformedMimeTypeException e) {

Log.e(TAG, "onResume=>error: ", e);

}

}

}

@Override

protected void onStop() {

PackageManager pm = getPackageManager();

pm.setComponentEnabledSetting(new ComponentName("com.bsm\_wqy.validationtools", "com.bsm\_wqy.validationtools.sensor.NFCTest"),

PackageManager.COMPONENT\_ENABLED\_STATE\_DISABLED, PackageManager.DONT\_KILL\_APP);

super.onStop();

}

@Override

protected void onDestroy() {

SystemProperties.set("persist.sys.nfctest", "false");

boolean isOpen = mNfcAdapter.isEnabled();

if (mLastNfcOpen != isOpen) {

if (mLastNfcOpen) {

mNfcAdapter.enable();

} else {

mNfcAdapter.disable();

}

}

super.onDestroy();

}

@Override

protected void onNewIntent(Intent intent) {

Log.d(TAG, "onNewIntent=>intent: " + intent);

if (intent != null) {

String action = intent.getAction();

if (NfcAdapter.ACTION\_TECH\_DISCOVERED.equals(action)

|| NfcAdapter.ACTION\_TAG\_DISCOVERED.equals(action)) {

mInfo.setTextSize(TypedValue.COMPLEX\_UNIT\_SP, 32);

mInfo.setTextColor(0xff00ff00);

mInfo.setText(R.string.pass);

if (mIsAutoTest) {

onClick(mPassButton);

}

}

}

}

}

5. 在AndroidManifest.xml文件中注册NFCTest.java类：

<!-- Xunhu: add NFC Test at 2016-11-02 09:44:06 by qty -->

<!-- Description: 添加NFC测试 -->

<activity

android:name=".sensor.NFCTest"

android:configChanges="keyboardHidden"

android:screenOrientation="portrait"

android:launchMode="singleInstance"

android:enabled="false" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.DEFUALT" />

</intent-filter>

<intent-filter>

<action android:name="android.nfc.action.TECH\_DISCOVERED" />

</intent-filter>

<meta-data android:name="android.nfc.action.TECH\_DISCOVERED"

android:resource="@xml/nfc\_tech\_filter" />

<intent-filter>

<action android:name="android.nfc.action.TAG\_DISCOVERED" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</activity>

<!-- Xunhu: end add NFC Test at 2016-08-10 11:24:08 by qty →

6. 启用NFCTest Activity：

///Xunhu: add NFC Test at 2016-08-10 11:24:08 by qty{{&&

///Description: 添加NFC测试

private void enabledNfcTestActivity() {

PackageManager pm = getPackageManager();

pm.setComponentEnabledSetting(new ComponentName("com.bsm\_wqy.validationtools", "com.bsm\_wqy.validationtools.sensor.NFCTest"),

PackageManager.COMPONENT\_ENABLED\_STATE\_ENABLED, PackageManager.DONT\_KILL\_APP);

}

///&&}}

7. 禁用NFCTest Activity:

PackageManager pm = getPackageManager();

pm.setComponentEnabledSetting(new ComponentName("com.bsm\_wqy.validationtools", "com.bsm\_wqy.validationtools.sensor.NFCTest"),

PackageManager.COMPONENT\_ENABLED\_STATE\_DISABLED, PackageManager.DONT\_KILL\_APP);