Step 1

Main Java widget

MyHostApduService

package de.androidcrypto.android\_hce\_beginner\_app;

import android.nfc.cardemulation.HostApduService;

import android.os.Bundle;

// add public class first, then extend and auto implement actions (processCommandApdu, onDeactivated.

public class MyHostApduService extends HostApduService {

@Override

public byte[] processCommandApdu(byte[] commandApdu, Bundle bundle) {

return new byte[0];

}

@Override

public void onDeactivated(int i) {

}

}

Step 2

AndroidManifest.xml

<activity

android:name=".MainActivity"

android:exported="true">

<intent-filter>

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

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

</intent-filter>

</activity>

=== Please add the following lines: ===

<service android:name=".MyHostApduService" android:exported="true"

android:permission="android.permission.BIND\_NFC\_SERVICE">

<intent-filter>

<action android:name="android.nfc.cardemulation.action.HOST\_APDU\_SERVICE"/>

<!-- category required!!! this was not included in official android HCE documentation -->

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

</intent-filter>

<meta-data android:name="android.nfc.cardemulation.host\_apdu\_service"

android:resource="@xml/apduservice"/>

</service>

=== end of copy and paste ===

</application>

...

service android:name: this is the name of your HCE service class you created in step 1 (“MyHostApduService”)

* android:permission: as our service class needs to get bound to an OS service we are doing it here
* intent-filter tag area: when the Android device is tapped to an NFC reader the device receives a communication request for a dedicated AID, here the parameters for the routing are defined
* action android:name: if the device receives a request the OS sends an intent to all apps using the “android.nfc.cardemulation.action.HOST\_APDU\_SERVICE” identifier to get notified about the request. Now it is up the each app to find out if the request (better the given AID) should be handled by this app
* category android:name: in the chapter “Verify that the HCE service is enabled” we saw that an Android device has two categories: “Payment” and “Other” and we need to define to what category this app or the service belongs. We are using the “Default” value.
* meta-data android:name: this is an accompanying information with the incoming intent and please use the name as it is — don not write the data in upper letters: “android.nfc.cardemulation.host\_apdu\_service”
* android:resource: the parameter is a XML file name that refers to more informations about e.g. AIDs in use by this app (“@xml/apduservice”) — see step 3

Step 3

XML Resource File

@xml/apduservice

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

<host-apdu-service xmlns:android="http://schemas.android.com/apk/res/android"

android:description="@string/servicedesc"

android:requireDeviceUnlock="false">

<aid-group android:description="@string/aiddescription"

android:category="other">

<!-- Sample for the demo application F2233445566 -->

<aid-filter android:name="@string/aidsimple"/>

</aid-group>

</host-apdu-service>

The <host-apdu-service> tag informs Android what the purpose of this file is

* android:description: is the name of the service (“HCE Service Beginner”)
* android:requireDeviceUnlock: is a security feature is the service should react on a request when the device is unlocked or not. I’m using the ”false” option, meaning the service will run even when the device is locked
* aid-group android:description: this entry refers to an entry in the strings.xml file (“HCE AID Description”)
* android:category: the refers again to the Payments or Others selection, we are using “other”
* aid-filter android:name: this refers again to the strings.xml file and the entry contains the (one or more) AID(s) the app is listen for: “F22334455667”