VTAP

Application Note –
Using the VTAP Android Host
Card Emulator example app

VTAP Core Firmware from v2.2.8.2
VTAP Connectivity Firmware from v1.0.4.3

VTAP50 and VTAP100

Revised August 2025 v1.0

DOT ORIGIN

If you need help to set up or use your VTAP reader, beyond what is contained in this Application Note, then please contact our support team.

Email: **vtap-support@dotorigin.com**

Download the latest documentation and firmware from **https://vtapnfc.com**

Telephone UK and Europe: +44 (0) 1428 685861

Telephone North America and Latin America: +1 (562) 262-9642

If you have any feedback on setting up or using your VTAP reader or this documentation, then please contact our support team. The product is constantly being reviewed and improved and we value feedback about your experience.

Contents

1   Host card emulation for Android passes

VTAP readers are designed to allow NFC Wallet passes, in Apple Wallet and Google Wallet, to deliver NFC payloads to a host system. NFC Wallet passes offer many advantages over using QR codes for this purpose - most significantly auto-selection and additional security features.

However, there are areas where it may not be possible to use a Google Wallet NFC pass, despite most Android phones having NFC enabled. For instance, there are many parts of the world where Google Wallet is not available either for payments or for Google Smart Tap NFC passes. (The list of countries where Google Wallet is currently supported can be found here **https://support.google.com/wallet/answer/12060037**).

In these cases, you could use the Android Host Card Emulation (HCE) feature from within your own Android app to provide similar NFC functionality for Android users, without using Google Wallet.

Android Host Card Emulation is a feature of the Android operating system that allows an app to emulate an NFC card. When this type of app is enabled and running as an Android service:

- The app registers its NFC application identifier (AID) with the Android operating system.

- The phone's NFC system will automatically route messages for this AID to the registered app, which can then respond appropriately.

This method can be used to easily send a payload (such as a user credential) from the app, via NFC, to a VTAP reader that is configured to request it, using the app's AID.

Dot Origin has produced a sample Android app to demonstrate how this works and will provide the source code project, to help Android developers to easily incorporate this feature into their own Android apps.

2 Develop an Android app to emulate an NFC Forum tag

There is an Android app example, which has been produced by Dot Origin, to demonstrate how an Android phone can interact, using NFC, to deliver a payload from the phone to a VTAP reader. Contact **vtap-support@dotorigin.com** for a copy of this example. This APK can be modified to build your own application.
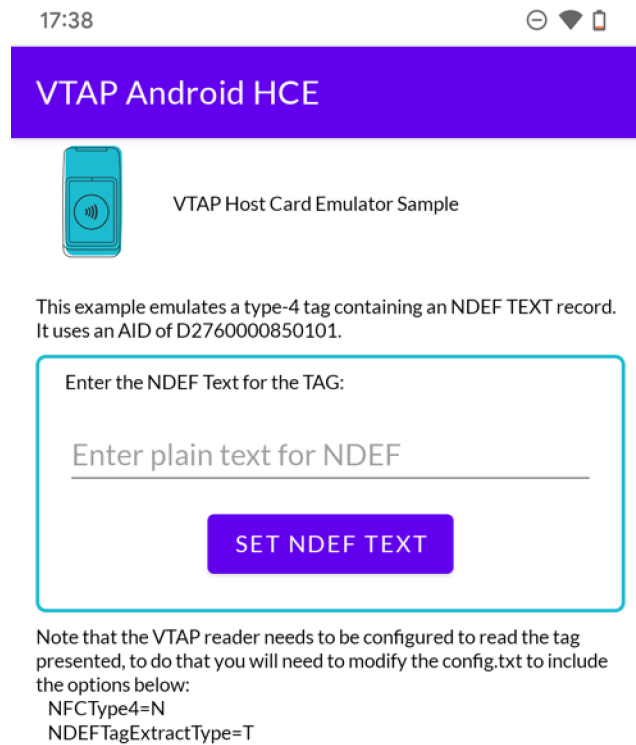

Figure 2-1 Example Android HCE app

When you receive the example, the pre-built APK file, which can be side-loaded to an Android phone, will be found in the `built apk` folder.

For more details about how it works, you can refer to the comments at the top of the file: `/app/src/main/java/com/dotorigin/vtapandroidhce/myHostApduService.java`

2.1 Choosing an AID

The application ID (AID) used by the Android HCE example app is a default NFC Forum NDEF AID: `D2760000850101`

An end-user's phone could have a number of host card emulation apps running, each using a different AID. So we recommend that, when you incorporate this functionality into your own app, your organisation should register and use a custom AID for your app. There is a registration system for AIDs here:
https://www.ansi.org/about/roles/registration-program/rid

Alternatively, ISO7816-5 reserves a range of AIDs for free use as unregistered custom AIDs. AIDs in this range start with `F0`. If you choose a custom AID in this range, at random, there is no guarantee that this will not conflict with another app using the same AID. However, the large number of possibilities in this range means that conflicts are unlikely.

When you have chosen your custom AID, you will need to change the AID used by the app. In the example code, the AID needs to be edited in two places:

- the AID defined in `myHostApduService.java`

  must match

- the `aid-filter` in `/app/src/main/res/xml/apduservice.xml`

You will also need to change the AID expected by the VTAP reader, away from its default. Use a setting `NDEFTagType4AID=F0123456123456` in the `config.txt` file on a VTAP reader, if `F0123456123456` was your chosen AID value.

3   Configure a VTAP reader to read tags from your app

Add the following lines to your VTAP `config.txt` file, to configure a VTAP reader to read the payload delivered by your app.

Example: Changes to `config.txt` to read tag data from your Android HCE app

```
!VTAPconfig

NFCType4=N
NDEFTagExtractType=T
NDEFTagType4AID=F0123456123456 ; replace F0123456123456 with your chosen AID
```

This configuration ensures that, in addition to reading mobile passes, the VTAP will receive NDEF Type 4 card/tag data, from any app with the set AID, and extract any NDEF record found in text format.

For more information on VTAP configuration options refer to the VTAP Commands Reference Guide, which can be obtained from **https://www.vtapnfc.com/documentation/**.

4 About Application Notes

Application Notes address topics of interest to small groups of customers, or topics around the use of a VTAP reader with third-party systems.

An Installation or Integration Guide for your VTAP reader model, together with the VTAP Configuration Guide, are the key documents to support the use of your VTAP reader. You will find the latest versions of these, and other useful information at **https://vtapnfc.com**.

If you need further help do contact us by email to **vtap-support@dotorigin.com**, or by phone +44 (0) 1428 685861 from Europe and Asia, or +1 (562) 262-9642 from Northern and Latin America.