OMNIKEY® ANDROID DRIVER
USER GUIDE

Software Version 1.9

PLT-02492, Rev. 1.5

February 2019

hidglobal.com

Copyright

Trademarks

Revision history

Date	Description	Revision
February 2019	Additional Android Studio information. Bluetooth removed. Appendices moved to section 1.	1.5
September 2016	New part number. OMNIKEY 5127 and OMNIKEY 5127 Mini readers added.	1.4
September 2012	Adaptation of Bluetooth description due to storage of MAC addresses in app preferences.	1.3
September 2012	Added notes on Android package installation from within another Android package.	1.2
September 2012	Added notes on system requirements, setup of development environment, Android package creation.	1.1
August 2012	Initial release.	1.0

Contacts

For additional offices around the world, see **www.hidglobal.com/contact/corporate-offices**.

Americas and Corporate	Asia Pacific
611 Center Ridge Drive Austin, TX 78753 USA Phone: +1 866 607 7339 Fax: +1 949 732 2120	19/F 625 King's Road North Point, Island East Hong Kong Phone: +852 3160 9833 Fax: +852 3160 4809
Europe, Middle East and Africa (EMEA)	**Brazil**
Haverhill Business Park, Phoenix Road Haverhill, Suffolk, CB9 7AE United Kingdom Phone: +44 (0) 1440 711 822 Fax: +44 (0) 1440 714 840	Condomínio Business Center Av. Ermano Marchetti, 1435 Galpão A2 - CEP 05038-001 Lapa - São Paulo / SP Brazil Phone: +55 11 5514-7100

HID Global Technical Support: **www.hidglobal.com/support**.

Contents

This page is intentionally left blank.

Section 1

1 Overview

1.1 Description

The OMNIKEY® Android driver package brings support for OMNIKEY smart card readers to the Android Operating System (OS). A management application, CardReaderManager, provides access to smart cards via the well-known JSR 268 API.

This document describes the usage of smart cards and readers under the Android OS, using the CardReaderManager management app, as well as smart card access via the JSR 268 Smart Card IO API (JSR 268 API).

1.2 Supported smart card readers

The following smart card readers are supported:

- smart@link chipset.
- OMNIKEY USB Readers: 1021, 3021, 3121, 3621, 3821, 5427, 5427 Gen2, 5022, 5023, 5025, 5122, 5122 Dual, 5127 Mini, 5422, 6121.

1.3 Minimum system requirements

- Android device (tablet or smart phone) with On The Go (OTG) support.
- Android OS version 4.4 or higher.
- API support for android.hardware.usb.

The apps can be installed without special privileges, but the user may have to grant access to USB devices during operation, if this is not already set.

1.4 Supported devices for USB smart card readers

Android tablets and smart phones which support the USB host (OTG) mode are supported. However, not all devices may offer a USB port to fit the USB Type A plug of the smart card readers. Please check what kind of adapter cables are needed to properly connect the reader to the devices. Due to inconsistencies between device manufacturers, adapters from micro-USB to USB are not always compatible between different vendors.

1.5 USB API test

To check whether an Android device supports USB host (OTG) mode, a free app, USB Device Info, can be downloaded from the Play Store. This app can be installed directly via the Play Store on the device, or downloaded directly from the Play Store web page: https://play.google.com/store/apps/details?id=aws.apps.usbDeviceEnumerator.

1.6 Definitions, abbreviations and symbols

APDU	Application Protocol Data Unit
API	Application Programming Interface
OID	Object Identifier
PC/SC	Personal Computer/Smart Card
IFD	Interface Device (for accessing ICC card)
OTG	On The Go, USB interface feature
OS	Operating System
SDK	Software Development Kit

1.7 References

ISO 7816-3	ISO 7816-3 Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols Third edition, 2006-11-01
ISO 7816-4	ISO 7816-4 Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for Interchange Second edition, 2005-01-15
PCSC-3	Interoperability Specification for ICCs and Personal Computer Systems Part 3. Requirements for PC-Connected Interface Devices Revision 2.01.09, June 2007
CCID	Specification for Integrated Circuit(s) Cards Interface Devices Revision 1.1, 22 April 2005
USB Dev Enum	https://play.google.com/store/apps/details?id=aws.apps.usbDeviceEnumerator
JSR 268 Doc	http://docs.oracle.com/javase/6/docs/jre/api/security/smartcardio/spec/javax/smartcardio/package-summary.html

Section 2

2 Installing Android packages (.apk)

To use the described features, it is first necessary to install the package on the Android device. This installation process includes the installation of a management app (CardReaderManager.apk), together with third-party apps that use the JSR 268 library for card reader communication.

The installation can be performed over USB or HTTP. This section will describe the steps necessary to complete the installation for both variants.

2.1 Installing over USB

When installing Android packages over a USB connection, the following prerequisites must be met:

- The Android SDK has to be installed on the host OS.
- A USB driver for the device has to be installed (Windows OS only).
- USB debugging has to be enabled on the device.

2.1.1 Android SDK

Independent of the OS used, the SDK for Android can be downloaded from **http://developer.android.com/sdk/index.html**.

After installation, the SDK Manager can be used to install the required packages. For package installation, the Android SDK Tools and Android SDK Platform tools have to be installed. Further, at least one of the platforms supporting API level greater or equal to 13 must be installed, which is necessary for Android development. To enable specific APIs, like Fragment or AsyncTaskLoader, the current version of the Android Support Library also has to be installed. For more information on package installation via the SDK Manager, see **http://developer.android.com/tools/extras/support-library.html**.

2.1.2 USB drivers (Windows OS only)

If the host OS is Windows Vista, 7 or 8, further USB drivers for the device are necessary in order to be able to establish the device connection over Android Debug Bridge (adb). The Android Developer website lists various vendors and links to their support pages where the drivers can be downloaded. Refer to **http://developer.android.com/tools/extras/oem-usb.html**.

2.1.3 USB debugging

On the Android device itself, USB debugging has to be enabled so that the device accepts the adb connection and grants the reception of commands, such as package removal and installation. The USB debugging configuration option is found in different places under different Android versions. Please refer to the specific instructions for your system version.

After the Android SDK, tools, and necessary USB drivers are installed, and USB debugging is activated on the device, it can be connected to the host with a USB device cable. The Android device will be automatically detected, which can be verified using the adb tool, which is part of the tools installed by the SDK Manager.

In a console (Linux or Windows) navigate to the folder where the Android SDK has been installed. The following example assumes a Windows OS and the installation path to be **C:\Users\<username>\AppData\Local\Android\Sdk\platform-tools**. The adb tool is installed in the platform-tools folder of the Android SDK. When called with the devices parameter, it shows all Android devices that are currently connected over USB.

```
C:\>cd Users\<username>\AppData\Local\Android\Sdk\platform-tools
C:\Users\<username>\AppData\Local\Android\Sdk\platform-tools> adb.exe devices
List of devices attached
0149CCCF0501300B device
```

When the device is shown in the above output, the USB connection has been set up successfully and Android packages can be installed and removed using adb. For this, the command line parameters `install` and `uninstall` can be used. With `install`, the path to the *.apk file has to be given, while `uninstall` takes the fully-qualified package name, for example:

```
C:\Users\<username>\AppData\Local\Android\Sdk\platform-tools> adb.exe install
"C:\Users\<username>\My Documents\app.apk"
860 KB/s (220167 bytes in 0.250s)
pkg: /data/local/tmp/app.apk
Success

C:\Users\<username>\AppData\Local\Android\Sdk\platform-tools> adb.exe uninstall
com.example.app
Success
```

Once installed, the Android device can be unplugged and the app can be used immediately on the device. See *Section 3 Using the management app*.

2.2 Installing over HTTP

An alternative installation option to USB debugging is to directly download and install the package on the Android device using HTTP access. A USB connection is not necessary for this procedure, but the following prerequisites must be met:

- The Android device must have access to the HTTP server where the Android package(s) are located.
- The user has to allow the installation of Android apps from sources other than the Play Store.

2.2.1 Allowing installation of Android apps from unknown sources

Android apps are normally installed via the Play Store, where the packages that are provided have passed certain security criteria. The Android security policy requires users to explicitly allow the installation of apps from unknown sources. The appropriate configuration option is found in different places for different Android versions. Please refer to the specific instructions for your version.

Once the **Unknown sources** option is enabled, the *.apk can be downloaded and installed

1. Download the Android package using the device's web browser.
2. Navigate to the **Download** folder and tap the downloaded package to begin installation. If the package has been previously installed, a notification message is shown.
3. The privileges required by the app are then displayed. Tap **Install** to accept the requirements and continue the installation.

2.2.2 Uninstalling apps on the device

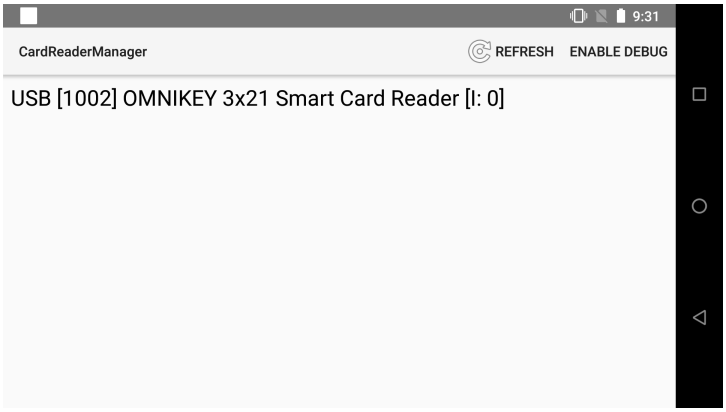
Please refer to the information specific to your version of Android, available online.

This page is intentionally left blank.

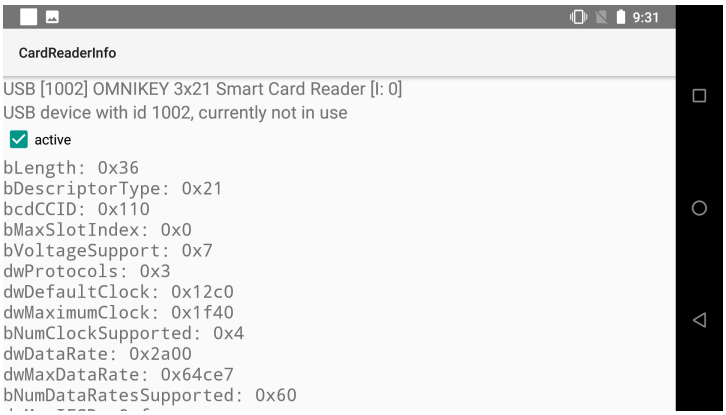
Section 3

3 Using the management app

When a supported USB reader is plugged into the Android device, a screen displays information about the connected USB device. To prevent this screen from displaying in the future, select the **Default** option. Tap **OK** to dismiss and launch the associated app which shows a list of currently managed devices. If the newly attached device is not listed, tap **REFRESH** to reload the reader list.



Further reader information (name, activity and usage status) is shown by tapping a reader name in the list. This screen is also used to activate or deactivate a reader. For example, if multiple readers are attached, and the user wants to expose specific reader to applications via the JSR 268 interface, the readers that are not needed can be deactivated in the reader information screen.



This page is intentionally left blank.

Section 4

4 Application development

Access to the readers and inserted smart cards is provided from the management tool to user apps via the public JSR 268 API. This API provides several classes and methods for reader listing, card presence detection, and data transmission. For further information, refer to: http://download.oracle.com/otndocs/jcp/jscio-4.0-fr-eval-oth-JSpec/.

4.1 Opening OMNIKEYDemo in Android Studio

1. In Android Studio, click **File** > **Open** and select the project root folder.
2. After launch, click **File** > **Sync Project with Gradle Files** to fetch all dependencies and configure the project.

   Note: Please use the OMNIKEYDemo application as an integration example.

4.2 Integrating the driver in a custom application

4.2.1 Requirements

There are two requirements to integrate the driver in a custom application running on an Android device:

- The CardReaderManager app must be installed on the device. It uses a service to provide access to the JSR 268 interface.
- The JSR268Library Android library must be included in the project as a dependency. Include library .aar in the libs folder and use the following code in build.gradle:

```
implementation files('libs/JSR268Library-release.aar')
```

Before integration, it is recommended to perform some additional steps.

First, to meet the first requirement above, the app should check if the CardReaderManager app is installed. To do this, a utility method from JSR268Library can be used (see code below).

Next, the app needs to declare a permission in **AndroidManifest.xml** to allow binding to the JSR268Library service, which allows reader access:

```
<uses-permission android:name="com.hidglobal.ia.omnikey.service.permission.SMARTCARDIO"/>
```

The app must handle the Android run-time permission request, which is obligatory for all devices running Android 6 and above. For convenience, the permission name constant is added in *android.smartcardio.hidglobal.Constants.PERMISSION_TO_BIND_BACKEND_SERVICE*.

To achieve this, the following code can be used:

```
// if CardReaderManager is installed bind card service from CardReaderManager app
if (PackageManagerQuery().isCardManagerAppInstalled(this)) {
  // check if app has permission to bind card service
  if (ContextCompat.checkSelfPermission(this, PERMISSION_TO_BIND_BACKEND_SERVICE) ==
  PackageManager.PERMISSION_GRANTED) {
    // bind service if permission granted - see code below
    bindCardService()
  } else {
    // request permission to bind service and expect result in onRequestPermissionsResult
    ActivityCompat.requestPermissions(this, arrayOf(PERMISSION_TO_BIND_BACKEND_SERVICE), REQUEST_
    BIND_BACKEND_SERVICE_PERMISSION)
  }
} else {
  // show dialog that CardReaderManager app is not installed
  showReaderAppNotInstalledDialog()
}
```

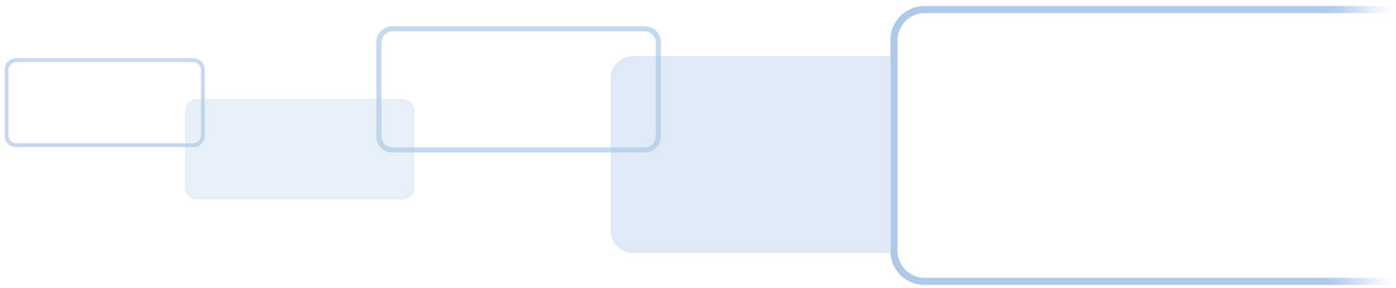
4.2.2 Integration

The app is now ready to integrate the driver. To gain access to the JSR interface, use the following code. Make sure that before execution, the user is granted permission to bind the service:

```
//bind to service (android.smartcardio.hidglobal.Constants.PERMISSION_TO_BIND_BACKEND_SERVICE
permission required)
CardService.getInstance(this, object : ServiceConnection {
  override fun onServiceConnected(componentName: ComponentName, iBinder: IBinder) {
    // service connected, JSR-268 available
    try {
      // use terminal factory as entry point to JSR-268
      // ex. list readers
      terminalFactory?.terminals()?.list()
    } catch (e: Exception) {
      // handle exceptions
    }
  }
override fun onServiceDisconnected(componentName: ComponentName) {
    // service disconnected
  }
}
```

Note: On Android devices running Android version below 6 (with no runtime permissions feature) the order of installing applications is important. Before installation of an application which integrates the driver, the CardReaderManager application must already be installed on the device. This is required because the CardReaderManager application defines the *android.smartcardio.hidglobal.Constants.PERMISSION_TO_BIND_BACKEND_SERVICE* custom permission and registers it in the Android system. If an app that attempts to use this permission is installed before the permission is registered, the system will not grant the permission to the app.

This page is intentionally left blank.



hidglobal.com