

# U.are.U® SDK v3

Platform Guide for Android

Copyright © 2015-2019 Crossmatch. All rights reserved. Specifications are subject to change without prior notice. The Crossmatch logo, Cross Match®, Crossmatch® and DigitalPersona® are trademarks or registered trademarks of Cross Match Technologies, Inc. in the United States and other countries. DigitalPersona®, TouchChip®, Eikon®, U.are.U® and FingerJet™ are trademarks or registered trademarks of DigitalPersona, Inc., which is owned by the parent company of Cross Match Technologies, Inc. All other brand and product names are trademarks or registered trademarks of their respective owners.

Published: April 18, 2019 (v3.2)

<b>INTRODUCTION</b>	<b>4</b>
<b>INSTALLATION</b>	<b>5</b>
System requirements.....	5
Development system .....	5
Target system (Android device) .....	5
Installation steps.....	5
Installing on the development system .....	5
Installing on the target system .....	6
<b>DEVELOPING APPLICATIONS WITH JAVA</b>	<b>7</b>
Prerequisites .....	7
The Java sample application .....	7
Getting and selecting a reader .....	8
Getting reader capabilities .....	9
Capture .....	10
Stream Image .....	11
Enrollment .....	12
Verification .....	13
Identification .....	14
<b>PAD: PRESENTATION ATTACK DETECTION</b>	<b>15</b>
Pre-Requisites .....	15
Enable/disable functions .....	15
False Accept/Reject functions.....	15
FAR/FRR relationships .....	16
<b>REDISTRIBUTION</b>	<b>17</b>
Redistributable documentation .....	17

THIS CHAPTER PROVIDES AN OVERVIEW OF THE CHAPTERS IN THE U.ARE.U SDK PLATFORM GUIDE FOR ANDROID-BASED PLATFORMS.

This guide describes how to use the U.are.U SDK to develop applications that will run on the Android platform. The U.are.U SDK is available for multiple platforms and this document describes information specific to developing applications that will run on Android-based platforms.

The guide contains the following chapters.

[Introduction](#) (this chapter) describes how to get the latest version of this documentation.

[Installation](#) provides instructions for installing on your development and target systems.

[Developing applications with Java](#) lists system requirements for developing and running applications using Java, provides additional installation instructions, and describes the included Java sample application.

NOTE: Native C/C++ development is possible, but requires root access in order to set permissions for the USB device. The native API is outlined in the "U.are.U Developers Guide", in the section "The C/C++ APIs." The sample code for the console application can be taken from the U.are.U SDK for Linux.

For a detailed description of the SDK, see the U.are.U SDK Developer Guide.

## Getting updated documentation

---

If you are viewing this guide from the U.are.U SDK download package, please check for the most current document on our website.

<https://www.crossmatch.com/company/support/documentation/>

THIS CHAPTER PROVIDES INSTRUCTIONS ON INSTALLING THE U.ARE.U SDK ON THE DEVELOPMENT AND TARGET SYSTEMS.

## System requirements

---

### Development system

---

The development system may be a Windows or Linux system. To develop Android applications, the Android SDK must be installed on the development system. The recommended IDE is Android Studio, version 3.0.1 or above.

### Target system (Android device)

---

The Android-based device must have Android 5.0 or above installed. A USB port in host mode is required. You don't have to have root privileges on the Android device.

The SDK works on a variety of hardware and is intended to have a small footprint so that it can run even on minimal hardware. Less capable hardware will work, but response time may not be optimal.

## Installation steps

---

The installation process is the same for all Android development, whether developing on a Linux or Windows-based machine.

There are two steps to the installation:

- Installing on the development system
- Installing on the target system

These steps are described below.

### Installing on the development system

---

#### Linux development system

To install the SDK on an Intel-based Linux development system:

1. Unpack `CM-UareU-3.2.0.xxx<date.build>.tar.gz` into a directory.
2. Go to the directory that you unpacked the file to.
3. Run `setup`.
4. Follow the onscreen instructions.

The installer copies all necessary files to the default or other specified folder. The files and folders described in the table below are installed on the development machine beneath the main product folder. The default is `<root>/crossmatch/urusdk-android/android/`.

Folder	Contents
bin/android-21	<ul style="list-style-type: none"> <li>• arm64-v8a</li> <li>• armeabi</li> <li>• x86</li> <li>• dpuareu.aar</li> <li>• dpuareu.jar</li> </ul>
docs	<ul style="list-style-type: none"> <li>• Java_API - Javadoc for Java API</li> <li>• EULA (End User License Agreement)</li> <li>• U.are.U SDK Developer Guide - describes all APIs</li> <li>• U.are.U SDK Platform Guide for Android - Android-specific details</li> <li>• Usage and Care Guides for DigitalPersona fingerprint readers or modules</li> </ul>
Samples/bin	Java executable for sample application: UareUSampleJava.apk
Samples/UareUSampleJava	Android Studio project files for the sample application

## Windows development system

1. Run DigitalPersona U.are.U SDK for Android.msi
2. Follow the onscreen instructions.

The installer copies all necessary files to the default or other specified folder. The files and folders are the same as those shown in the table above and are installed on the development machine beneath the main product folder. The default (for 64-bit) is C:\Program Files (x86)\DigitalPersona\U.are.U SDK\Android\ and for 32-bit, C:\Program Files\DigitalPersona\U.are.U SDK\Android\.

## Installing on the target system

To install the run-time environment on the target hardware platform:

1. Enable USB debugging mode by navigating on your device's display to Settings->DeveloperOptions->Enable USB debugging.
2. Connect the target hardware platform to the development system.
3. Establish an adb connection with the device. For example,

```
adb install <UareU SDK folder>/urusdk-android/Android/Samples/bin/areUSampleJava.apk
```

Installing the .apk file will place the following components into a separate application zone:

- Driver and engine libraries
- Java SDK wrapper
- Java Sample Application

If you install any of those components elsewhere on the target, you will have duplicate files installed. The intended architecture of Android is for all applications to have all dependencies secured within their own application zones. All dependencies, UIs and services are installed and started automatically.

# Developing applications with Java

THIS CHAPTER PROVIDES INFORMATION NECESSARY FOR DEVELOPING ANDROID APPLICATIONS IN JAVA WITH THE U.ARE.U SDK.

## Prerequisites

---

This chapter assumes that you have a working knowledge of Java and that you know how to develop for Linux readers.

## The Java sample application

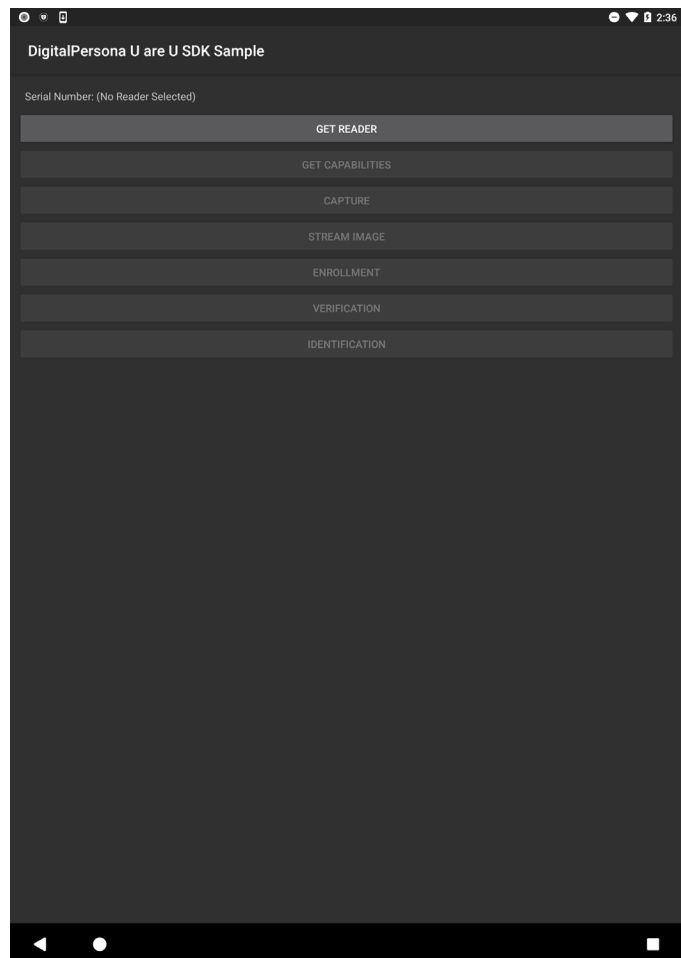
---

The U.are.U SDK for Android includes a sample application to demonstrate the features of the SDK when using the Java API. The sample application is located in the `Samples` folder.

When you launch the application, you see the main screen as shown below.

The sample program demonstrates:

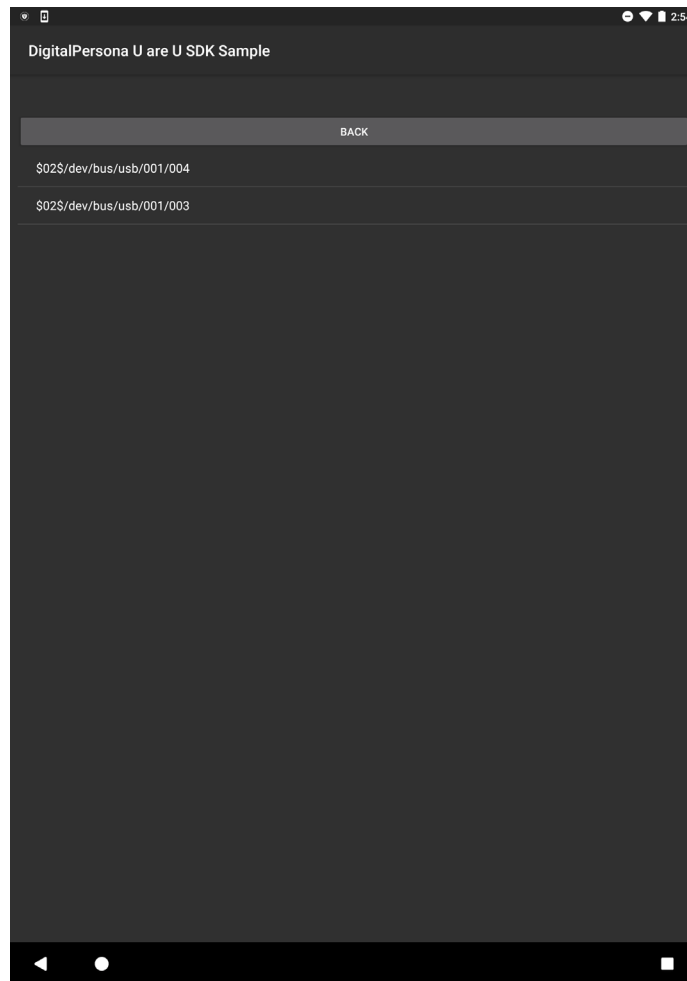
- Getting and selecting a reader
- Getting a reader's capabilities
- Capturing a fingerprint
- Live streaming a fingerprint image
- Enrollment
- Verification
- Identification



## Getting and selecting a reader

---

To get identifying information about any readers available to the program, click the **GET READER** button. If there is only one reader attached, that reader will be selected automatically. If there are multiple readers, you will see a list of available readers and you can select the desired device, as shown below.



Click on the desired device and then click the **Back** button.

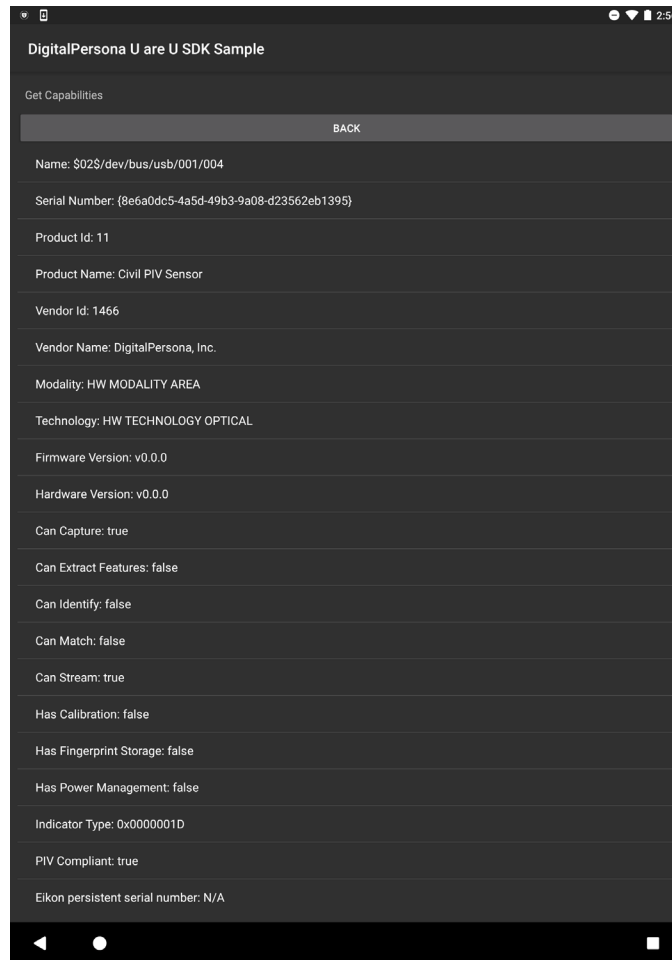


## Getting reader capabilities

---

To get the reader capabilities, click the **GET CAPABILITIES** button.

The capabilities will be displayed, as shown in the image on the right.



Click on the **Back** button to return to the main screen.

## Capture

To demonstrate capturing a fingerprint image, click the **Capture** button. This puts the reader into capture mode and an image of the captured fingerprint is displayed in the window.



To exit capture mode, click on **Back**.

## Stream Image

---

To demonstrate streaming a fingerprint image, click the **Stream Image** button.

This puts the reader into streaming mode, where the window becomes like a live window on the reader as it streams the results. Placing a finger on the reader displays the streamed fingerprint, and removing the finger shows a blank screen.



To exit streaming mode, click on **Back**.

## Enrollment

---

Click on **Enrollment** to begin enrolling a test subject.

You will see a series of prompts to scan fingers for enrollment, as shown below.

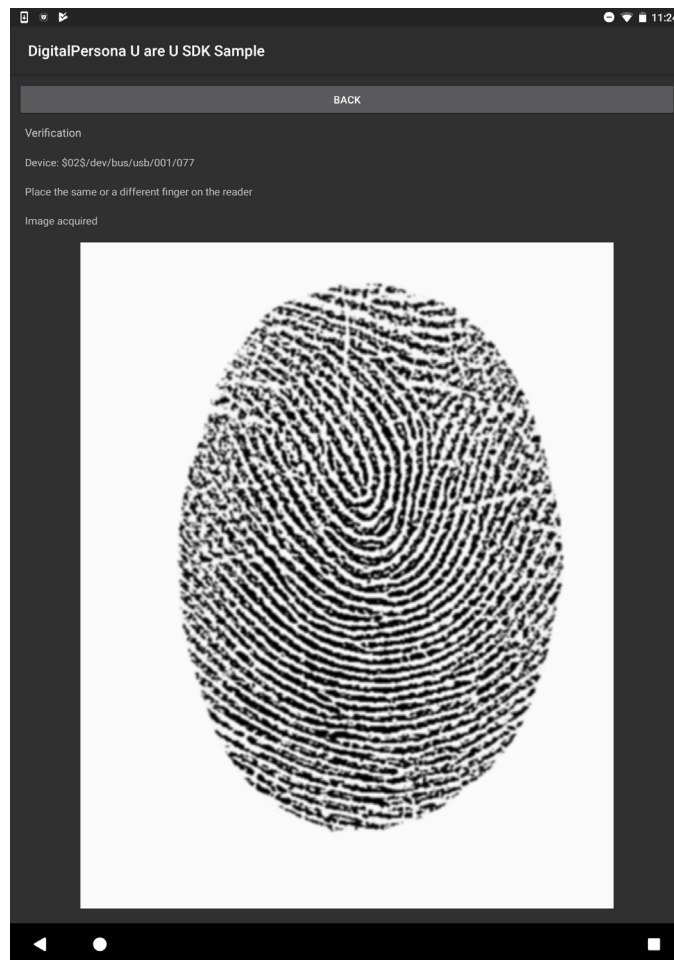
After the first finger is successfully scanned, you will be prompted to scan additional fingers until a sufficient number of high quality scans are complete. The number of fingers requested will vary depending on the image scans.



The enrollment function will continue to request scans until an acceptable enrollment record has been created. When enrollment is complete, click **Back** to return to the main screen.

## Verification

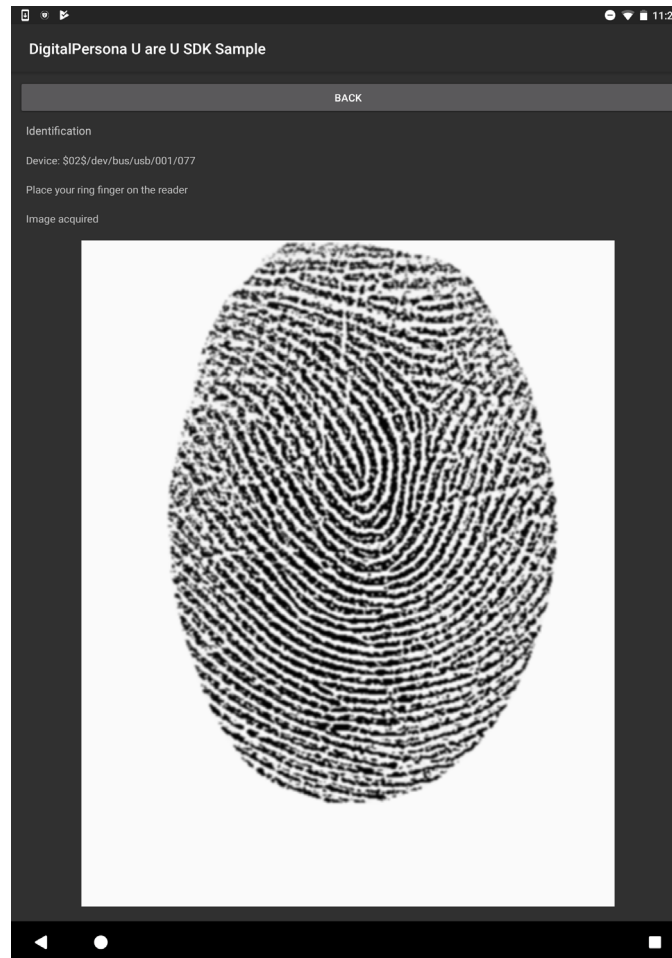
To test the verification feature, click on the **Verification** button. Verification is a 1-to-1 comparison where the application matches against a specified fingerprint. When you click the **Verification** button, you will be prompted to place any finger on the reader, which will create a fingerprint template to match against. Next you will be prompted to place either the same finger or a different finger on the reader.



If you place the same finger on the reader, you will get a match. Using a different finger will give a no match result. To exit verification mode, click on the **Back** button.

## Identification

To test the Identification feature, click the **Identification** button. Identification is a 1-to-many comparison where the application searches through a collection of enrolled fingers to find a match. The sample application prompts you to place various fingers on the reader (your thumb, index, middle and index fingers) so that it can create an FMD.



After an FMD is created, you will be prompted to place any finger on the reader to identify against the FMD that was just created. If you place a previously enrolled finger on the reader, a match will be indicated and the above enrollment sequence will begin again.

To exit identification mode, click on the **Back** button.

# PAD: Presentation Attack Detection

THIS CHAPTER DESCRIBES HOW TO USE THE PRESENTATION ATTACK DETECTION (PAD) FEATURE OF THE U.ARE.U SDK.

## Pre-Requisites

---

When installing without the DigitalPersona Authentication Service, the U.are.U Android SDK (version 3.2.0 and above) includes a Presentation Attack Detection (PAD) option against 3-dimensional (molded) spoofs which can be used with U.are.U 5300 readers and modules only.

By default, the PAD algorithm is turned off, but it can be turned on by calling `dpfpdd_set_parameter()` with the `parm_id` equal to `DPFPDD_PARMID_PAD_ENABLE` with the first byte of the buffer set to 1.

*Regardless of whether or not the 3-dimensional PAD algorithm is enabled, the DigitalPersona 2-dimensional (printed) spoof detection is always enable for U.are.U 5300 readers and modules.*

## Enable/disable functions

---

//enable PAD

```
unsigned char bEnable = 1;
```

```
dpfpdd_set_parameter(hReader, DPFPDD_PARMID_PAD_ENABLE, 1, &bEnable);
```

//disable PAD

```
unsigned char bEnable = 0;
```

```
dpfpdd_set_parameter(hReader, DPFPDD_PARMID_PAD_ENABLE, 1, &bEnable);
```

//check if PAD is enabled

```
unsigned char bEnabled = 0;
```

```
dpfpdd_get_parameter(hReader, DPFPDD_PARMID_PAD_ENABLE, 1, &bEnabled);
```

```
if(bEnabled) {
```

```
    //PAD enabled
```

```
}
```

```
else {
```

```
    //PAD disabled
```

```
}
```

## False Accept/Reject functions

---

The default False Accept Rate is 3%. The target False Accept Rate for PAD can be changed by calling `dpfpdd_set_parameter()` with `parm_id` equal to `DPFPDD_PAD_CONFIDENCE` with the first byte of the buffer set to 1.

//set FAR at 5%

```
unsigned char bFAR = 5;
```

```
dpfpdd_set_parameter(hReader, DFPD_PAD_CONFIDENCE, 1, &bFAR);
```

```
//get current FAR
```

```
unsigned char bFAR = 0;
```

```
dpfpdd_get_parameter(hReader, DFPD_PAD_CONFIDENCE, 1, &bFAR);
```

When a spoofed finger is detected, the quality field of the DFPD\_CAPTURE\_RESULT structure returned from dpfpdd\_capture() will have a value of DFPD\_QUALITY\_FAKE\_FINGER.

## FAR/FRR relationships

---

Setting the False Accept Rate determines the corresponding False Reject Rate, as shown in the table below.

False Accept Rate (FAR) %	False Reject Rate (FRR) %
49	0.087
47	0.087
46	0.092
43	0.114
41	0.119
39	0.126
37	0.143
34	0.153
32	0.165
31	0.180
28	0.211
27	0.233
25	0.243
24	0.265
22	0.311
21	0.325
18	0.403
17	0.454
16	0.478
15	0.544
14	0.609
13	0.638
12	0.704
11	0.804
10	0.961
9	1.073
8	1.258
7	1.430
6	1.629
5	1.937
4	2.304
3	2.957
2	4.208
1	8.049



THIS CHAPTER PROVIDES DETAILS ABOUT THE FILES INCLUDED IN THE U.are.U SDK THAT MUST BE DISTRIBUTED WITH YOUR APPLICATION.

## Redistributable documentation

---

You may redistribute the end-user documentation included in the software package's Redist directory to your users, according to the terms of the product EULA, which is located in the following directory after installation (Windows) or unpacking (Linux).

- On Windows - [Installation location]\DigitalPersona\U.are.U SDK\Android\docs
- On Linux - [Installation location]/Crossmatch/urusdk-android/Android/docs

The documentation contained in the Redist directory includes Regulatory and Warranty information as well as Usage and Care Guides for DigitalPersona and TouchChip readers or modules that should be provided to anyone purchasing readers or modules from you.