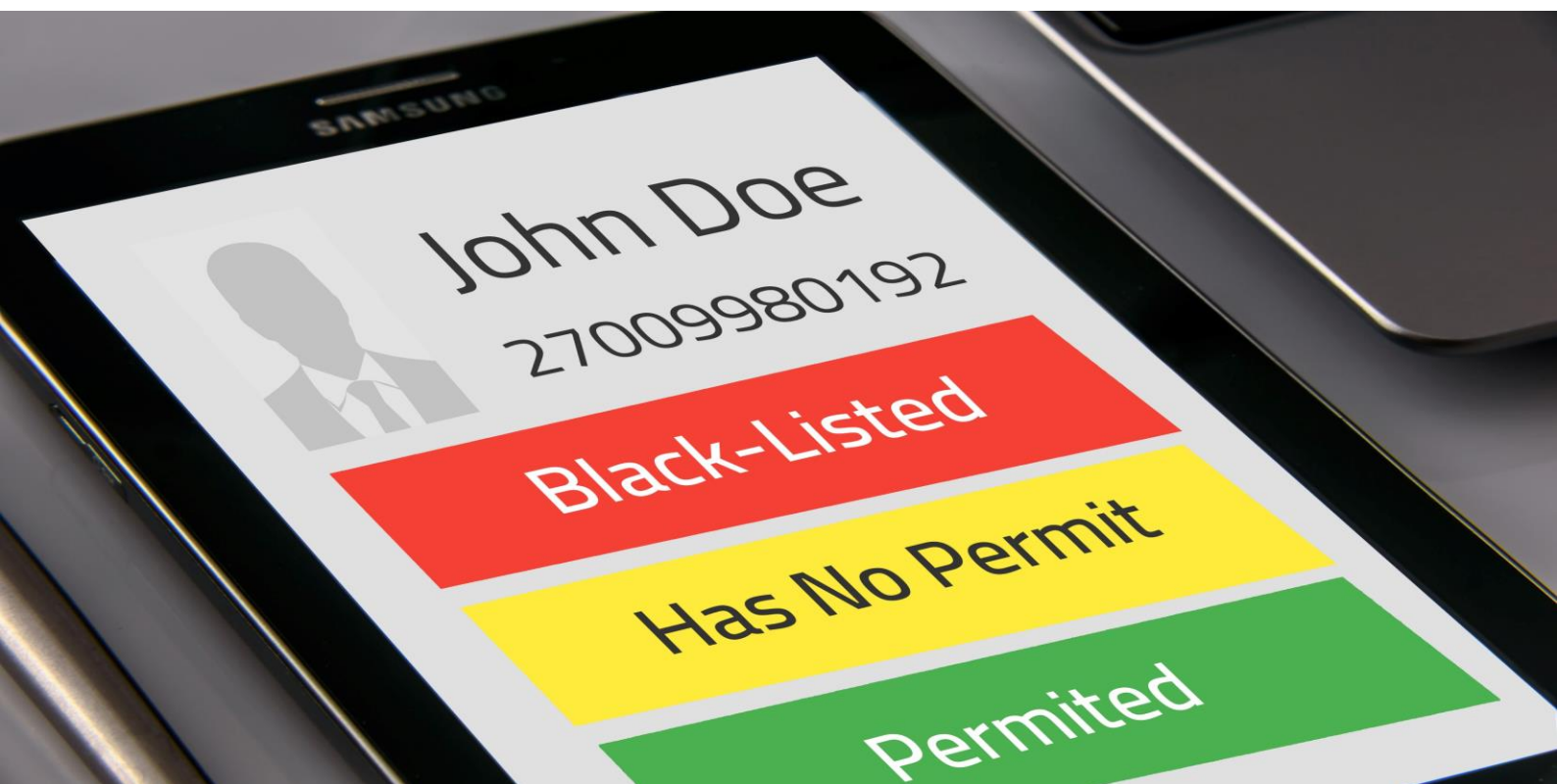


Sharper Software Guides

Smart Civil ID Library for Android Developer's Guide

“A computer shall not waste your time or require you to do more work than is strictly necessary.”

Jef Raskin



Prepared by:
Hagop Karaguezian
Co-Founder
Sharper software



Sharper Software Guides

TABLE OF CONTENTS

Software Development Kit (SDK)	3
Target Audience	3
Platform Requirements	3
General Walkthrough	3
Sample Application	4
Class Reference	4
KwCivilldReader	4
Method 'create'	4
Method 'end'	4
Method 'setOnDataAvailableListener'	4
OnKwCivilldDataAvailableListener	4
KwCivilldDataPayload	5
Method 'read'	5
KwCivilldData	5
Fields and Child Objects	5
Exceptions	7
InvalidCivilldException	7
ReadCivilldFailedException	8

Sharper Software Guides

SOFTWARE DEVELOPMENT KIT (SDK)

The Kuwait Smart Civil ID Reader SDK by Sharper Software enables developers to integrate the Kuwait Smart Civil ID into their Android applications using the ACR38U-ND device.

Contents of SDK:

1. Libraries (kwcivilid.jar and acssmc-1.1.2.jar) to be included as dependencies in the target application.
2. Sample project for Android Studio by Google.
3. This documentation.

TARGET AUDIENCE

This document is intended for application developers on the Android platform. Guidance is given on how to develop applications using the SDK primarily on Android Studio by Google.

PLATFORM REQUIREMENTS

The SDK is targeted toward Android version 4.3 or later. It is not supported under older OS versions.

GENERAL WALKTHROUGH

For Android Studio projects, you can drop the two JAR files (kwcivilid.jar and acssmc-1.1.2.jar) into the libs folder under the project module's folder. Otherwise, a dependency should be taken on the JAR files, and import the namespace `com.sharpersoftware.kwcivilid`.

The SDK's main entry point is the `KwCivilIdReader` class. This class should be treated as a singleton and handled from a central location.

To obtain a reference to the `KwCivilIdReader` class you can call the static method `create` and passing in an instance of an `Activity` as the host. The host is used during initialization to perform the following actions:

1. Ask for USB device permission to access the smart card reader.
2. Initialize the reader device and prepare for detecting inserted cards.

After the following steps are completed, the initialization `Activity` is no longer held by the SDK.

In order to receive notifications when a card is inserted, you need to register by providing a callback to the `setOnDataAvailableListener` instance method. Only one callback is ever active at any time.

Once notification is received regarding a new card being inserted, you can proceed to read the data. An instance of `KwCivilIdDataPayload` is passed to the callback, which can be used to call the `read` instance method to receive the data. It is highly recommended to call `read` inside of an asynchronous task (`AsyncTask` or similar).

After reading has completed, an instance of `KwCivilIdData` is obtained. This instance holds all the Civil ID information in memory and accessible through various fields, which will be explained further later on.

Sharper Software Guides

SAMPLE APPLICATION

A sample application is provided that performs the minimum steps necessary to read from the smart Civil ID.

Take note of the following key points:

1. Only one instance is ever created of `KwCivilldReader`. This instance is kept alive as long the application is running (whether active or in the background).
2. Whenever the Activity is created (or recreated), a listener is attached using `setOnDataAvailableListener`. This allows the continued operation of the reader even when the activity is recreated, due to – for example – screen rotation.
3. The activity of reading from the card is performed in an `AsyncTask`. This allows you to keep the UI responsive while reading the card.
 - a. You may even choose to show a busy indicator during this process. This is not demonstrated in the sample.
4. You can keep a reference to the instance of `KwCivilldData` for as long as necessary.
 - a. Once the read operation is completed, the instance is no longer tracked by the reader.
 - b. Calling read again will result in a new instance of `KwCivilldData` with the same actual data. This assumes the card has not been removed since the original read operation.

CLASS REFERENCE

`KwCivilldReader`

This is the main entry point to the SDK. Of notable interest is the 'create' static method which creates an instance of the reader.

Method 'create'

A static method that requires a single parameter of type 'Activity'. The activity is used to initialize the USB reader device and gain permission.

Method 'end'

An instance method that closes all handles to the reader and invalidates the `KwCivilldReader` instance. After this method has been called the reader is no longer active and you will not receive notifications about detected cards.

Method 'setOnDataAvailableListener'

An instance method that registers a callback of type 'OnKwCivilldDataAvailableListener'. This callback is called when a new and valid Kuwait Civil ID card is detected.

`OnKwCivilldDataAvailableListener`

A callback interface which requires the implementation of the single method 'onKwCivilldDataAvailable' which delivers an instance of 'KwCivilldDataPayload'.

Sharper Software Guides

KwCivilIdDataPayload

A delegate for the card which allows reading the actual Civil ID data. At the point of receiving an instance of this type, the smart card is initialized but not fully read. You can call the 'read' instance method to obtain the data stored on the card.

Method 'read'

An instance method that reads the data from the smart card. This step can take anywhere between 1 to 7 seconds, depending on various factors. Therefore, it is highly recommended to perform this action in an asynchronous task (AsyncTask or similar).

KwCivilIdData

This class is a read-only data carrier that stores all the data from the Civil ID smart card. All data is in-memory and no further interaction occurs with the smart card while accessing any of the fields in this class.

You can also safely keep hold of references to instances of this class as the SDK itself no longer uses it.

The class is also serializable in case you want to persist the information.

Fields and Child Objects

Field / Path	Type	Data	Notes
civilId	String	12-digit unique Civil ID number	
arabicName.full	String	The full name of the Civil ID holder in Arabic	
arabicName.prefix	String	Honorific, such as the title of H.H. the Emir of Kuwait, and other dignitaries.	Usually blank.
arabicName.first	String	First name in Arabic.	
arabicName.middle	String	Middle name in Arabic.	
arabicName.last	String	Last name in Arabic.	
arabicName.suffix	String	Honorific.	Usually blank.
englishName.full	String	The full name of the Civil ID holder in English (or using Latin alphabet characters).	
englishName.suffix	String	Blank	Not used.
englishName.first	String	First name in English.	
englishName.middle	String	Middle name in English.	

Sharper Software Guides

Field / Path	Type	Data	Notes
englishName.last	String	Last name in English.	
englishName.suffix	String	Blank	Not used.
sex	Enum:Sex	Sex of the card holder.	
dateOfBirth	KwDateOfBirthInfo	The date of birth of the card holder.	Some older citizens and residents only have the year of birth designated. Use the 'year', 'month' and 'day' fields for these instances, or use 'date' which may return a date like 1/1/1950 for a card holder whose date of birth is known only in year.
nationality	Enum:Country	The nationality of the card holder.	The Country enumeration type has textual information that you can use to format the text either as country name or as nationality adjective, e.g. "Kuwait" vs. "Kuwaiti".
bloodType	Enum:BloodType	The blood type of the card holder.	Usually accurate, but may be missing for some very young card holders.
address.uniqueKey	String	The unique key for this address.	This is usually a number that can uniquely identify an address in the database of the Public Authority for Civil Information. This number can also be directly searched for in the Kuwait Finder app developed by PACI.
address.district	String	The name of the area or district (Arabic only).	
address.block	String	Block, usually a number.	
address.street	String	Street name or number.	
address.building	String	Building, usually a number.	
address.unitType	String	Arabic text describing the unit type, such as Villa, Flat, etc.	There is no English translation for this field, as the domain of the data changes frequently.
address.unitNumber	String	The number of the flat, for example.	Sometimes blank.
address.floor	String	Usually floor number.	Sometimes blank.
card.serialNumber	String	The serial number of the current card.	Mostly numeric, this is not useful to uniquely identify a person, as the card number changes with every physical card, even for the same person.

Sharper Software Guides

Field / Path	Type	Data	Notes
card.issueDate	Date	The date the card was issued.	
card.expiryDate	Date	The date the card expired or will expire on.	Usually corresponds to the date the residence permit expires for residents.
contacts.email	String	The email address of the card holder.	Unreliable, as most people do not register their email address with PACI.
contacts.getMobile()	KwPhoneInfo	The mobile number of the card holder.	Unreliable, as the information may be missing or out of date.
contacts.getWork()	KwPhoneInfo	The work phone number of the card holder.	Unreliable, as the information may be missing or out of date.
contacts.getHome()	KwPhoneInfo	The home phone number of the card holder.	Unreliable, as the information may be missing or out of date.
residence.article	String	The article of law under which the residence permit was issued for the resident.	Arabic only. Non-Kuwaitis only.
residence.sponsor	String	Name of the sponsor (person or organization) of the residence permit.	Arabic only. Non-Kuwaitis only.
documentNumber	String	Internal for PACI use.	Has limited application outside of PACI.
moiReference	String	Unused; formerly for use of Ministry of Interior.	
moiReferenceIndication	String	Unused; formerly for use of Ministry of Interior.	
guardianCivilId	String	The Civil ID number of the legal guardian of the card holder.	Applies only in case the card holder is a dependent.

Exceptions

InvalidCivilIdException

This exception is thrown when the inserted card is not a Civil ID, or is otherwise an unrecognizable smart card.

This exception should be treated as if the inserted card is not a Civil ID. You should not attempt to retry the read operation.

Sharper Software Guides

ReadCivildFailedException

This exception is thrown in case the read operation fails for any reason. This could be due to either hardware or software failure.

This exception indicates the possibility that a retry may yield successful results.

Either of these exceptions could occur if the Civil ID smart card is damaged, depending on the type of damage sustained by the card.

Copyright Notice and Disclaimer

THIS INSTRUCTIONS MANUAL CONSISTING OF A TOTAL OF 8 PAGES INCLUSIVE OF THIS PAGE IS THE INTELLECTUAL PROPERTY OF SHARPER SOFTWARE ([HTTP://WWW.SHARPERSOFTWARE.COM](http://www.sharpersoftware.com)) AND IS LICENSED TO YOU FOR THE PURPOSES OF THIS TRAINING ONLY. IT IS PROTECTED BY KUWAITI LAW NO. 63 FOR YEAR 1999 REGARDING INTELLECTUAL PROPERTY RIGHTS. IT IS ILLEGAL TO MAKE COPIES OF THIS DOCUMENT OR THE ACCOMPANYING SOFTWARE, FULLY OR IN PART WITHOUT A WRITTEN PERMISSION FROM SHARPER SOFTWARE. FOR MORE INFORMATION ABOUT THIS LAW, PLEASE VISIT <http://www.wipo.int/wipolex/en/details.jsp?id=2784>.

SHARPER SOFTWARE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

THE EXAMPLE COMPANIES, ORGANIZATIONS, PRODUCTS, DOMAIN NAMES, E-MAIL ADDRESSES, LOGOS, PEOPLE, PLACES, AND EVENTS DEPICTED HEREIN ARE COMPLETELY FICTITIOUS. NO ASSOCIATION WITH ANY REAL COMPANY, ORGANIZATION, PRODUCT, DOMAIN NAME, E-MAIL ADDRESS, LOGO, PERSON, PLACE, OR EVENT IS INTENDED OR SHOULD BE INFERRED.

Copyright © 2011-2017 Sharper Software. All rights reserved under Kuwaiti Law #64/1999.